

Opis organizacji obozów informatycznych w Szczecinie

Tomasz Nowak

Ten dokument wyjaśnia sposób organizacji obozów informatycznych w Szczecinie (przygotowujących do Olimpiady Informatycznej). Przedstawione techniki są oparte na wieloletnim doświadczeniu uczniów / studentów i pozwalają uniknąć większości błędów oraz znacząco skrócić czas potrzebny na zorganizowanie takiego obozu.

Dokument jest publicznie dostępny pod adresem tonowak.com/oi.

Dokument został napisany pod czytelników będącymi uczniami / absolwentami Szczecina, jednak zamieszczone informacje mogą się przydać innym miastom (i dlatego ten dokument został upubliczniony), a wtedy trzeba mieć na uwadze to, że część założeń może nie być prawdziwa (może zmienić się rozmiar kadry, możliwości ośrodka, poziom grup uczestników, czas trwania obozu, ilość czasu potrzebnego na zorganizowanie obozu).

Wszelkie poprawki do dokumentu są mile widziane. Można zgłaszać prośby o dodanie rozdziału lub można też podzielić się doświadczeniem z organizacją obozów w innych miastach. Najchętniej przyjmuję poprawki Merge Requestem na GitLabie gitlab.com/tonowak/oi-pdfs (o ile ktoś jest na tyle odważny, aby nauczyć się to zrobić), ale chętnie też przyjmuję mailowo na adres tomasz.nowak@tonowak.com.

1 Plan pracy

Materiały na obóz da się całkowicie przygotować ściśle przed obozem (i rzeczywiście udawały nam się takie obozy, w których wszystkie rundy były przygotowane przed wyjazdem). Jednak zazwyczaj się tak nie udaje i trzeba przygotowywać w danym dniu rundy na kolejne dni obozu — takich sytuacji należy, oczywiście, unikać, gdyż oznacza to niewyspane noce, spadek jakości treści i testów lub mniej czasu na relaks.

Dokument przedstawi w jaki sposób zaplanować pracę i jak podzielić ją między osoby przygotowujące obóz.

Zakładane jest, że wśród organizatorów obozu znajduje się:

- nauczyciel — osoba zajmująca się sprawami organizacyjnymi, to znaczy załatwieniem ośrodka, jedzenia, transportu oraz zajmująca się formalnościami,
- być może inni opiekunowie, których zadaniem jest tylko zwiększenie liczby osób, które mogą być na obozie (zazwyczaj nauczyciele lub uczniowie z uprawnieniami opiekuna (które można sobie wyrobić w tydzień lub dwa tygodnie prawie całkowicie zdalnie)),
- od dwóch do pięciu osób będących w kadrze zadaniowej — osoby zajmujące się przygotowaniem zadań, contestów.

2 Podział uczestników na grupy

Stosujemy następujący podział uczestników na grupy (nazwy nawiązują do dywizji na platformie zadaniowej Codeforces):

- div1 — uczestnicy, którzy systematycznie w ciągu roku trenują, zostali finalistami / laureatami lub jest duża szansa, że zostaną (z zamysłu ma to być mało liczna grupa),

- div2 — licealiści przygotowujący się do drugiego etapu, niekoniecznie systematyczni, nie powinni mieć problemów z zadaniami z OIJ. **Nie należy się przejmować tym, że niektóre osoby są w div2 już któryś raz. Celem tej dywizji jest przygotowanie osób, które chcą być przygotowane do drugiego etapu i systematycznie pracują.**
- div3 — uczniowie z podstawówki lub początkujący licealiści, znają podstawy programowania, poziom zadań z OIJ,
- div4 — osoby nieumiejące programować lub umiejące w małym stopniu, jeszcze nie potrafią rozwiązywać zadań z OIJ.

Przygotowanie obozu należy zacząć od podziału uczestników na powyższe grupy, albo chociażby należy oszacować licznosci grup i ich precyzyjniejszego poziomu (grupy celowo mają nieściśłą definicję poziomu uczestników, gdyż i tak jest to trudne do szacowania w rzeczywistości).

Bardzo częstym błędem jest przeszacowanie umiejętności uczestników. Przy każdym obozie zdarza się, że dana grupa posiada mniejsze umiejętności, niż było to oczekiwane przez kadrę zadaniową. Dobrze jest więc szacować w dół umiejętności zawodników, gdyż szacowanie w górę nigdy nie kończy się dobrze. Dobrym pomysłem jest przeprowadzenie testu kwalifikacyjnego na około miesiąc przed obozem. Nawet jeśli napisze go niewielka część uczestników, zawsze jest to jakaś informacja o poziomie grupy. Szacowanie poziomu na podstawie listy uczestników zazwyczaj nie kończy się dobrze, gdyż umiejętności uczestników potrafią zarówno znacząco wzrosnąć jak i, o dziwo, *zmaleć* pomiędzy obozami.

Zdarza się, że podzbiór tych grup tworzy zbiór pusty i to nie jest nic złego. Wtedy nadal używamy takich nazw grup. Na przykład może istnieć obóz z grupami „div2” oraz „div4”, ale bez grup „div1” oraz „div3”. Wydaje się to być mylącym wyborem nazw, jednak kadra zadaniowa poprzez nawiązania do nazw grup na Codeforcach od razu wie, jakiego poziomu zawodników się spodziewać, a także jest to pomocne dla uczestników, gdy przychodzą na kolejne takie obozy (uczestnik wie, że nie powinien trafić do grupy o większym numerze, niż na poprzednim obozie).

3 SIM

W Szczecinie do wrzucania zadań i tworzenia contestów korzystamy z systemu o nazwie SIM. System ten został przygotowany przez ucznia liceum, aktualnie w niedużym stopniu jest wciąż przez tę osobę utrzymywany.

SIM to nasz Szczeciński skarb. W przeciwieństwie do sio2, Satori oraz solve, nasz SIM po prostu działa bez niczyjej pomocy. System sio2 istnieje do prowadzenia różnych konkursów na dużą skalę, w tym Olimpiady Informatycznej, a Satori do sprawdzania zadań nie tylko algorytmicznych (i chwala tym systemom, że dobrze się sprawują do swojej roli), zaś SIM istnieje tylko i wyłącznie po to, aby wygodnie wrzucać na niego zadania oraz łatwo tworzyć testy / rundy. Został dobrze przetestowany, jest praktycznie bezproblemowy i niesamowicie szybki w porównaniu do jakiegokolwiek innej platformy. Zachwycamy się SIM'em dniami i nocami.

SIM'a można bardzo łatwo zainstalować. Dzięki temu tworzymy oddzielny serwer obozowy ze świeżutkim SIM'em, z którego korzystamy podczas obozu (niezależnie od głównego serwera SIM'owego, który jest używany w szkole), a następnie mergujemy go po obozie z głównym SIM'em (co też jest łatwe, dzięki licznym, dobrze przygotowanym skryptom).

Dodatkowo pod SIM'a zostało stworzone proste, wyspecjalizowane narzędzie do tworzenia zadań. To narzędzie (o nazwie SIP, wspomnimy o nim też później) symuluje działanie SIM'a podczas testowania paczek zadań, dlatego można sobie spokojnie tworzyć zadanie na swoim komputerze i ma się pewność, że po wrzuceniu stworzonej paczki na SIM'a, ta paczka po prostu będzie działała.

Na obóz zabieramy ze sobą jeden komputer, który spełnia rolę serwera z SIM'em, oraz router / routery. Tworzymy na obozie lokalną sieć, do której uczestnicy podłączają się aby korzystać z SIM'a. Ma to takie zalety, że nie trzeba mieć dostępu do internetu, więc wszystko śmiga niezależnie od warunków ośrodka.

Instrukcje instalacji SIM'a znajdują się [tutaj](#) (czyli na GitHubie, gdzie też należy składać uwagi do działania lub funkcjonalności SIM'a (poprzez Issues)), a instrukcje ekspresowej instalacji znajdują się [tutaj](#).

4 Przed obozem

Ściśle przed obozem należy wykonać jak najwięcej przygotowań. Pozwala to doszlifować zadania i uniknąć większości problemów podczas obozu. W to wlicza się:

- wyznaczenie grup obozu,
- uzbieranie odpowiedniej ilości zadań,
- wybór z listy zadań i podział na contesty,
- przygotowanie paczek,
- doszlifowanie paczek (sprawdzanie treści, itp)
- przygotowanie lokalnego serwera SIM
- stworzenie listy i zebranie potrzebnych rzeczy na obóz (na przykład tablice)

5 Zbieranie zadań

Używamy liniowej skali oceniania trudności zadań, ściśle związanej z trudnościami zadań na Codeforcach. Dla wygody, każde zadanie oceniamy od 1 do 1, co odpowiada Codeforce'owym trudnościom od 8 do 26, z przeskokiem 2 co każdą trudność.

Na Codeforcach, zadanie ma trudność X , gdy osoba o ratingu X ma 5% szans na rozwiązanie tego zadania podczas Codeforcowego contestu (oraz osoba o ratingu $X - 2$ ma około 25% szans). Daje to więc dobre szacowanie na to, ile średnio osób rozwiąże dane zadanie. Przy czym należy pamiętać, że przydzielona trudność zadania na Codeforcach jest przy założeniu, że zadanie pojawiło się na 2 – 2.5-godzinnym conteste z kilkoma innymi zadaniami, więc zadanie na Codeforcach o trudności X ma mniejszą trudność w contestach 5-godzinnych z trzema zadaniami, dlatego przy zadaniach z Codeforców należy zaniżyć ich trudność.

Na obozy **nie używamy autorskich zadań**. Autorskie zadania powinny trafić na konkursy (którym zawsze brakuje sensownych, oryginalnych zadań), nie na obozy. Czyli, należy rozważyć zaproponowanie zadań na Olimpiadę Informatyczną, AMPPZ, a w dalszej kolejności zaproponowanie bardziej lokalnym konkursom lub platformom zadaniowym typu Codeforces, AtCoder. Większość tych jednostek zapłaci za zaproponowanie zadania. Dopiero, gdy (bezowocnie) podjęło się próby zaproponowania zadania w różne miejsca (a jeżeli zadanie jest oryginalne i sensowne, nie powinno się to zdarzyć), można myśleć o użyciu zadania na obozie.

Dlatego na obozie używamy praktycznie tylko *ukradzionych zadań*, czyli zadań z różnych konkursów lub platform zadaniowych, zazwyczaj ze zmienioną/przetłumaczoną treścią (co zapobiega googlowania kodów i rozwiązań przez zawodników) i, jeżeli jest to możliwe, takimi samymi testami i rozwiązaniami.

Jest drobna kwestia prawna oraz moralna – czy można kraść zadania? Jest to ciężki (zazwyczaj przemilczany) temat, a szczególnie, że do prawie wszystkich zadań nie ma zapewnionej licencji. Spotkałem się z praktyką, aby po zawodach umieszczać informację o źródle zadań, co powinno rozwiązać wszelkie kwestie. Ja osobiście podchodzę do tego w taki sposób, że dopóki ukradzione zadania nie pojawiają się na zawodach, a są używane tylko i wyłącznie w celach treningowych, to ukradnięcie nie jest złe – zawodnik zawsze mógłby zasubmitować na oryginalnej platformie, albo pobrać testy z danego źródła i uruchomić rozwiązanie lokalnie, a ukradnięcie tylko sprawia, że zawodnik może bardziej się skupić na samym treningu (i też zawodnik nie jest kuszony sprawdzeniem rozwiązania, jeżeli nie wie gdzie go szukać). Zresztą, opublikowanie testów samo w sobie niebezpośrednio jest zgodną na kradzież.

Zbieranie zadań sprowadza się więc do znania dobrych miejsc do kradzieży (zdefiniowane jako: źródła prawdopodobnie nieznanymi zawodnikom, które mają zapewnione treści zadań, rozwiązania, testy oraz opcjonalnie omówienia, dzięki czemu jest mało potrzebnej pracy, aby przygotować paczkę). W dokumencie nie będą umieszczone informacje o tym, jakie źródła uważamy za dobre (jeden powód jest taki, że przez to może się zmienić kryterium „prawdopodobnie nieznanymi zawodnikom”, przez co nie będą już „dobre”), ale będą umieszczone informacje o tym, jak wygodnie przerobić takie zadania na gotowe paczki.

Jak najwcześniej (to znaczy kilka miesięcy przed, albo nawet zazwyczaj rok przed) obozem tworzymy arkusz z zadaniami, które mogłyby się nadać na jakiś obóz. W ciągu roku, gdy piszemy contesty i napotkamy jakieś ciekawsze zadanie, wpisujemy je na listę. Każde zadanie ma zapisany tytuł, link do źródła, informację co jest już zapewnione przez źródło (to znaczy testy, treści, kody), krótki opis rozwiązania / potrzebnych technik oraz ocenę trudności. Jeżeli lista będzie systematycznie wypełniana zadaniami z różnorodnych źródeł i ze zróżnicowanym poziomem trudności (niekoniecznie tylko zadaniami, które mają już gotowe testy i kody – bardzo często warto spędzić czas na stworzenie pełnej paczki ciekawego zadania, jeżeli okaże się wyjątkowo edukacyjne dla uczestników), to przed obozem lista powinna być wystarczająco długa.

Jeżeli przed obozem okaże się, że zadań jest za mało, zaczynamy szukać pasujących zadań z różnych platform. Najbardziej w ciągu roku na listę trafiają łatwe zadania (o trudnościach 1 – 5), ale na szczęście można je łatwo znaleźć, gdy się wie, gdzie ich szukać. Trochę gorzej jest ze znalezieniem średnio-trudnych zadań, które zawierają podzadania. Często kończy się to dodawaniem sztucznych podzadań do zadań zerojedynkowych (w stylu ICPC), byleby zadania wyglądały tak, jakby były stworzone z myślą o 5-godzinnych contestach. To oczywiście nie jest dobre rozwiązanie, dlatego od samego początku powinno się mieć na uwadze źródła, które dobrze wpasowują się w formę contestów.

6 Wyznaczenie contestów

Mając gotową listę potencjalnych zadań na obóz, należy wybrać część z nich, które zostaną użyte na contestach. Do tego konieczna jest wiedza o licznosci i poziomie grup na obozie, liczbie contestów i ich czasie trwania.

Pierwsze dwie kwestie zostały już poruszone, pozostałe dwie zostaną poruszone w dalszych częściach dokumentu.

Dla $div1$, rozkład trudności zadań każdego dnia mógłby wyglądać tak, że są dwa średnie zadania, na przykład o trudnościach 6-8 (lub jedno łatwiejsze, drugie średnie), których sumaryczna liczba setek powinna być niezerowa (najlepiej coś pomiędzy .5, a 1.5 razy licznosc grupy) oraz jedno trudne zadanie (na przykład o trudności 9-1), którego prawdopodobnie nikt nie zrobi (co ma na celu zapełnienie czasu zawodnikom, którym udałoby się zrobić pozostałe dwa zadania). W skrócie – ma być ostro dla zawodników. To jest już grupa, która najwięcej uczy się na omówieniach i dobijaniu po wyzerowaniu zawodów, a nie grupa, która ma przećwiczyć implementację lub szybkość myślenia.

Im grupa ma większy numer (to znaczy kolejno $div2$, $div3$, $div4$), tym łagodniej powinno być. Dlatego na $div2$ można dać jedno łatwe (na przykład trudność 4-6) i dwa średnie zadania (na przykład trudności 6-8), zaś dla $div3$ dwa łatwe i jedno średnie (na przykład trudności kolejno 2-4, 3-5, 4-6).

Można zauważyć, że przecięcie trudności zadań pomiędzy dwoma kolejnymi grupami jest niezerowe. Pozwala to na wykorzystanie strategii, aby dwie sąsiednie grupy miały jedno wspólne zadanie na contestach. Jest to dobry sposób zmniejszenia czasu potrzebnego na przygotowanie obozu, jednak jeżeli się da, lepiej tego uniknąć – osoby z różnych grup siedzące obok siebie mogłyby zauważyć, że mają wspólne zadanie, przez co wiedzą, które jest łatwiejsze / trudniejsze, co niszczy umiejętność oceniania trudności zadań na zawodach.

Dodatkowo, dla $div1$ kluczowe jest, aby każdego dnia pojawiło się pewne *okropne* zadanie, które jest albo trudne implementacyjnie, albo trudne koncepcyjnie (a najlepiej to i to). Dzięki temu nauczą się rozwiązywać nieprzyjemne zadania i będą mieli co robić, gdyby zbliżali się do dobiecia wszystkich zadań po zawodach (dlatego też dobrze jest dać trudne implementacyjnie zadania na początku obozu, aby od samego początku mieli czym się zająć wieczorami).

Na podstawie opisów rozwiązań i źródeł można wybrać na dany contest w miarę różnorodne zadania. Należy też się postarać, aby na obozie nie pojawiły się dwa zadania o podobnych rozwiązaniach.

Nieuzyte zadania, które zostały na liście, można po prostu wykorzystać na przyszłych obozach (czyli jest tylko jedna lista zadań do wykorzystania, która z czasem się powiększa, a na obozach znacząco zmniejsza).

7 Przygotowywanie paczek

SIP (narzędzie do tworzenia paczek pod SIM'a) ma pewne własności i feature'y, które czynią proces przygotowywania paczek efektywnym i wyspecjalizowanym.

Przede wszystkim, uruchomienie SIP'a na paczce symuluje działanie SIM'a, to znaczy uruchamia rozwiązania wzorcowe w taki sam sposób, z tym samym sandbox'em. Oznacza to, że upload paczki na SIM'a jest przewidywalny i bez niespodzianek. Szczególnie, jeżeli korzysta się z domyślnej opcji automatycznego ustawiania limitów czasowych zadania na podstawie rozwiązania wzorcowego (mogą nastąpić drobne wahania w punktacji rozwiązań przez różny stosunek prędkości działania programów, ale występuje to zdecydowanie rzadziej, niż częściej), dzięki czemu skutecznie można pozbyć się stałych limitów czasu. Z tego powodu nie śpieszy się nam z uploadem paczek na SIM'a, byleby lokalnie działały. Daje to pewien spory plus – nie trzeba mieć dostępu do lokalnego serwera obozowego, żeby skończyć przygotowanie paczki, więc można to zrobić długo przed obozem lub w miejscu bez internetu/dostępu do serwera (drobna uwaga: trzeba tylko pamiętać o tym, by wrzucić te zadania przed contestem. ..)

Osobiście bardzo unikam momentu, w którym mam pobrać paczkę z serwera obozowego i ją zmienić (by poprawić coś w treści, testach lub rozwiązaniach). Trzeba wtedy pobrać całą paczkę, znaleźć ją w tym folderze Downloads, bawić się w unzip'owanie jej i przejście do jej odpowiednich folderów. Najgorsze jest to, że nie ma to kontroli wersji. Co gdy inna osoba w tym samym czasie też zedytuje paczkę i zrobi upload? Jest spora szansa, że nikt nie zauważy, że zmiany zostały stracone. Często też próbuje się unikać pobierania paczki z serwera obozowego i pracuje się na lokalnej wersji paczki, która też może nie być aktualna. Same kłopoty, których często nikt nie zauważa.

Nasz aktualny system wygląda tak, że korzystamy z git'a (na przykład na GitLab'ie) do kontroli wersji wszystkich paczek z jednego obozu. W przypadku, gdy na obozie jest słaby dostęp do internetu (czego, swoją drogą, należy unikać), to można przerzucić repozytorium na serwer git'a postawiony na serwerze obozowym.

Wymaga to względnie małych rozmiarów paczek na gicie, przez co nie uploadujemy testów, binarek ani pdf'ów do repozytorium. Zamiast tego są tylko pliki .tex, rozwiązania, generatorki i skrypty. Dopiero przy konieczności uploadu paczki na SIM'a lub sprawdzenia rozwiązań, lokalnie uruchamiamy komendę tworzącą testy na podstawie configu i generatorki (ale i tak odpowiedni plik *.gitignore* nie wrzuca tych plików na repozytorium).

Co z ukradzionymi paczkami, które mają już testy? Do takich źródeł w miarę możliwości trzymamy skrypty, które na podstawie nazwy zadania lub linku do testów pobierają te testy i odpowiednio je grupują. Wtedy taki skrypt jest uruchamiany zamiast generatorki (choć wciąż tą samą komendą). Ma to ten minus, że potrzebny jest internet do „wygenerowania” testów, ale cóż.

Podążając tą ścieżką, do kilku źródeł z paczkami stworzyliśmy skrypty, które na podstawie nazwy zadania tworzą pełną paczkę z testami (o ile są dostępne), rozwiązaniami (o ile są dostępne) i treścią (o ile istnieje). Niedawno do naszego arsenału doszedł skrypt do tłumaczenia zadań z angielskiego na polski, który jeszcze bardziej usprawnia proces ukradania paczek. W niektórych przypadkach ukradnięcie paczki to kwestia uruchomienia jednej prostej komendy. Przy czym drobna uwaga – podczas pisania tych wszelkich skryptów bardzo kusi, aby napisać je byle jak, czyli by po prostu działały na kilku przykładach. Ale przecież cały sens polega na tym, że pisanie skryptów kradnących zadania to inwestycja w zaoszczędzony czas na kolejnych obozach, więc warto postarać się napisać te skrypty porządnie i dokumentować je (nawet, jeżeli osobiście tych skryptów się już więcej nie użyje, to wciąż przyszłe pokolenia mogłyby zyskać na korzystaniu z nich).

Poradnik do tworzenia paczek używając SIP'a znajduje się [tutaj](#).

Weryfikacji zadań nie mamy bardzo rozwiniętej. Po tym, gdy dana osoba skończy tworzyć paczkę, to jakaś inna osoba ma za zadanie sprawdzić treść (w co wlicza się użycie programu **aspell**, który szuka literówek w słowach (i pomija część kodu odpowiadającą za wyrażenia matematyczne) za pomocą słownika, dzięki czemu można bardzo szybko można poprawić wszelkie literówki w treści), przejrzeć rozwiązania i sprawdzić na oko, czy testy są w tej formie, co jest opisane w treści (często na porządnych konkursach ten proces jest zastępowany *weryfikatorką*, czyli programem, który sprawdza, czy testy dokładnie spełniają wszystkie warunki, to znaczy ograniczenia oraz białe znaki, ale my to pomijamy dla zaoszczędzenia czasu).

Aktualny stan przygotowywania paczek utrzymujemy w arkuszu, aby wiedzieć jakie paczki są gotowe, a jakie jeszcze wymagają stworzenia / zweryfikowania.

8 Przygotowanie sprzętu

Przed obozem dobrze jest przygotować lokalny serwer SIM, czyli zainstalować na jakimś komputerze (nie ma większego znaczenia na jakim, a jak nie ma żadnego stacjonarnego, to laptop pewnie też styknie) jakiegoś na przykład Debian'a i na nim postawić SIM'a. Nieważne, czy się świeżo zainstalowało SIM'a, czy się już posiadało jakiegoś, dobrze jest też sprawdzić, czy działają sprawdzaczki oraz czy działają wszystkie języki podczas submitowania.

Po wieloletnich zabawach z rozstawianiem dużej ilości routerów w różne zakątki ośrodka zauważyliśmy, że tak naprawdę wystarczy jeden wysokiej jakości router, aby każdy mógł na conteeście się podłączyć do sieci obozowej oraz aby sieć była dostępna w pokojach uczestników. Wystarczy więc przygotować jeden, dobry router z dwoma niedługimi kablami Ethernet (około 6 metrów).

Po przygotowaniu serwera i routera, dobrze jest zasymulować sieć obozową i sprawdzić przed wyjazdem, czy można swobodnie podłączyć się do SIM'a. Ale i tak na wszelki wypadek dobrze jest wziąć jakiś zapasowy router oraz klawiaturę i myszkę, by się podłączyć do serwera. Nie zaszkodzi też zapasowy komputer, który mógł by służyć jako serwer.

Do omówień i wykładów potrzebne są dwie tablice, najlepiej białe i duże. Koniecznie trzeba wziąć dwie tablice, bo inaczej nie można zrównoległać omówień i wykładów różnych grup. Do tego oczywiście należy dorzucić ścierki i pisaki (sporo czarnych i jak najwięcej kolorowych).

Treści na drużynówki / mecz trzeba albo wydrukować przed obozem, albo wziąć ze sobą drukarkę.

Dla uczestników też trzeba zabrać jak najwięcej przedłużaczy (by je odpowiednio rozstawić w sali contestowej, aby każdy laptop miał dostęp do prądu podczas zawodów). Jak się da, to można też wziąć zapasowe laptopy dla uczestników. Dobrze jest też przed obozem poprosić uczestników, aby zabrali ze sobą przedłużacze (co najwyżej nadmiarowe przedłużacze wykorzystają u siebie w pokojach).

Zazwyczaj w ośrodkach jest rzutnik lub telewizor, który można użyć podczas pewnych wykładów, a zazwyczaj taki sprzęt jest na kabel VGA. Dobrze jest więc wziąć taki kabel VGA oraz przejściówkę do HDMI.

Dodatkowe dobre klawiatury i myszki są mile widziane, ale dodatkowe monitory nie są potrzebne.

Poza tym, można jeszcze wziąć podstawowe narzędzia typu śrubokręty, nożyczki, taśmy, pendrive'y.

Często w ośrodkach nie ma zapewnionego czajnika, więc można przynieść swój, z odpowiednimi zapasami herbaty (oraz po przyjeździe na obóz koniecznie należy zapewnić wystarczająco dużą ilość butelek wody mineralnej dla uczestników).

Nauczyciel powinien zadbać też o rzeczy higieniczne typu apteczka, termometry, maseczki, płyny do dezynfekcji rąk oraz powierzchni.

W czasie wolnym są mile widziane jakieś piłki, badminton, albo nawet puzzle. Z doświadczenia, rozłożenie 1000 puzzli na stole dostępnym dla uczestników pierwszego dnia obozu jest dobrym pomysłem. Chociaż wskazane jest zaopatrzyć się w takie, które mają więcej niż 1000 elementów, bo uczestnicy zaskakująco szybko je kończą. (Może kiedyś uda się załatwić customowe puzzle?)

Dobrze jest przygotować arkusz z rzeczami, które należy zabrać na obóz, i w nim odznaczać jakie rzeczy są przygotowane przed obozem, a następnie jakie rzeczy już wrzucono do autokaru.

9 Ośrodek

Kwestia głównie pozostawiona nauczycielowi.

Dopowiem tylko, że miejsce nie ma większego znaczenia, byle miało zapewnione jedzenie oraz przynajmniej jedną salę dostępną dla uczestników 24/7 z wystarczającą liczbą miejsc siedzących, a najlepiej jeszcze dodatkową, drugą salę dla omówień i wykładów.

Dobrze jest jak najwcześniej ustalić z właścicielami ośrodka godziny posiłków (zanim inne grupy w ośrodku sobie zaklepią swoje godziny). Koniecznie musi być pięć godzin przerwy między końcem śniadania, a początkiem obiadu. Zwlekanie z ustaleniem godzin posiłków może skutkować skróceniem czasów contestów, co zmniejsza ich podobieństwo do prawdziwych zawodów.

Można też spróbować załatwić jakieś pączki i drożdżówki podczas contestów (pięć godzin od śniadania do obiadu to jednak niemało), choć to już spoza ośrodka.

10 Plan obozu

Po planowanym przyjeździe do ośrodka dobrze jest przeznaczyć jedną godzinę dla uczestników (i dla kadry), by się rozpakować. Jeżeli jest sporo czasu, to można też wyjść z uczestnikami do pobliskiego sklepu (ale jest to zaskakująco czasochłonna, więc również ryzykowna, opcja). Wtedy też można poprosić uczestników, aby nieśli ze sobą wodę mineralną do ośrodka (na koszt obozu, aby była dostępna dla wszystkich).

Rozpoczęcie obozu to tak naprawdę półgodzinna sesja wyjaśniania uczestnikom, aby podłączyli ładowarki do przedłużaczy, podłączyli się do sieci Wi-Fi, zarejestrowali się na SIM'ie i zasubmitowali na zadanie próbne (które może być po prostu zadaniem na zsumowanie long longów). Kluczową czynnością jest też zabronienie uczestnikom siedzenia obok osoby ze swojej klasy – znacząco ogranicza to ilość rozmów podczas zawodów (przy czym nie trzeba planować kto gdzie ma siedzieć, bo najwyraźniej wystarczy powiedzieć zawodnikom, że nie mają siedzieć obok osoby ze swojej klasy i to działa całkiem dobrze).

Dobrze jest tego samego dnia spróbować wcisnąć 5-godzinny contest po obiedzie z omówieniem po kolacji (zamiast wykładu, który zazwyczaj jest po kolacji).

Standardowy dzień to contest po śniadaniu, a po obiedzie mieszanka czasu wolnego, omówienia i dobijania (przy czym dobijanie to tak naprawdę też teoretycznie czas wolny i szczerze mówiąc nie wiem, czy wpiśnięcie tego do harmonogramu jakkolwiek pomaga z ilością dobijania u zawodników). Po kolacji trochę czasu wolnego i po tym wykład.

Jest kilka wyjątków – ognisko jakiegoś wieczoru (czyli skrócony wykład), zaplanowane opcjonalne wyjście do kościoła w niedzielę (najlepiej po południu, ale to zależy od miasta, w którym jest ośrodek) oraz ostatni dzień, który głównie ma mecz dla div1 oraz div2 (zaczynający się przedostatniego dnia) oraz drużynówkę dla div3 oraz div4.

Dobrze jest umieścić plan obozu w miejscu dobrze widocznym dla uczestników. Zazwyczaj planujemy cały obóz i umieszczamy harmonogram pierwszego dnia (wtedy, gdy jest sesja logowania się do SIM'a). Każdy uczestnik powinien zawsze wiedzieć kiedy i gdzie ma być.

11 Contesty

Głównie kierować się zasadą „div1 ostro, div4 najłagodniej”. Czyli div1 standardowy contest na zasadach drugiego lub trzeciego etapu OI, div2 podobnie (ale łagodniejsze zadania), zaś div3 już contesty z odsłonięciem punktów (lub całkowitymi odsłonięciami), być może contestami 4-godzinnymi lub być może contesty z krótką przerwą na podpowiedzi. Contesty w formacie OI okazują się nie być najlepszą metodą na przygotowanie do OIJ, dlatego dla div3 warto przygotować contesty, które mają więcej prostszych zadań (na przykład 5 zamiast 3), aby jak najlepiej przećwiczyć implementację i szybkość myślenia. Ponadto takie rozwiązanie, przy odpowiednim doborze zadań, sprawia, że div3 nadaje się również dla uczestników zupełnie początkujących. Dzięki temu można w pewien sposób zastąpić istnienie czwartej dywizji, której zorganizowanie często jest niemożliwe ze względu na zbyt małą liczbę osób w kadrze.

Dla div4 niestety trzeba zorganizować zupełnie oddzielną formę contestów (i dlatego należy unikać posiadania tej dywizji na obozie, bo jest najbardziej czasochłonna), gdzie mają pełne odsłonięcia, wykład o danym typie zadań przed contestem, wyjaśnianie błędów i być może rozwiązań, czyli w skrócie – mają mieć warsztaty, a nie contest.

Podczas contestu dobrze jest zapewnić pewną przekąskę (na przykład drożdżówki i pączki), ale wtedy trzeba uważać na to, aby przerwa nie była zbyt długa i głośna. Należy też sprawić, aby uczestnicy nie bali się podchodzić do kadry zadaniowej w celu zgłaszania problemów z contestem lub w przypadku pytań do treści.

Tak poza tym, jeżeli contest jest dobrze przygotowany, to dla kadry to jest to tak naprawdę czas na relaks lub na odespanie.

12 Wykłady i omówienia

W skrócie – na temat. Czyli unikać wykładania na rzeczy niezwiązane z Olimpiadą Informatyczną, lub o tematach za trudnych, aby się pojawiły na takich konkursach. Poza tym należy mieć na uwadze to, że koniecznie trzeba się w niezerowym stopniu przygotować do tych omówień i wykładów, bo pełne improwizacje zazwyczaj nie kończą się idealnie. W ramach przygotowań dobrze jest spróbować oszacować długość wykładu / omówienia i zaplanować co skrócić, jeżeli wykład / omówienie zajmie zbyt dużo czasu (a praktycznie zawsze wychodzi dłuższe, niż planowane).

Uczestnicy zazwyczaj szybko zapominają treść wykładu, jeżeli nie dostają żadnych zadań do zaimplementowania z tego wykładu. Dlatego kluczowe jest, aby **po wykładzie umieścić zadania o tej tematyce**. Takie zadania nazywamy „powykładowymi” lub „warsztatowymi” (druga nazwa bierze się z „warsztatów”, w których kadra pomaga uczestnikom rozwiązać zadania). Przynajmniej jedno zadanie powinno być po prostu na zaimplementowanie treści wykładu (czyli często zadanie na jakąś „technikę”) oraz przynajmniej jedno zadanie powinno wymagać pomyślenia w jaki sposób wykorzystać materiał wykładu (i kodu z pierwszego zadania). Często zdarza się, że uczestnicy dobrze znają potrzebny materiał i trenują głównie umiejętność rozwiązywania zadań. Wówczas warto jest udostępnić zadania przed wykładem, a nie po, aby ci, którzy już znają omawiane zagadnienia mogli je przeciwżyć bez marnowania czasu.

Podczas prowadzenia wykładów dla dywizji div3 oraz div4 trzeba mieć na uwadze, że słuchacze często nie znają odpowiednich podstaw (na przykład wykłady o dynamikach na wstępie powinny mieć totalnie trywialne zadania, zamiast od razu przechodzić do średnich) oraz praktycznie zawsze nie przerywają wykładu, gdy przestają go rozumieć (więc trzeba często się upewniać, że nadążają z treścią wykładu).

Wykłady dla div1 z reguły nie mają sensu, bo członkowie tej dywizji mają już dobrze opanowane techniki potrzebne do startu w finale OI. Zamiast tego, można przeprowadzić warsztaty, na których z uczestnikami rozwiązuje się zadania przy tablicy. Najlepiej jak zadania przygotowane na warsztaty są w podobnej tematyce i mają stopniowo rosnący poziom trudności. Należy jednak unikać zadań bardzo trudnych (na poziomie JOI lub IOI), gdyż jednego wieczoru nie starcza czasu, żeby odpowiednio długo nad nimi pomyśleć. Coś pomiędzy 4 a 6 zadań na dwie-trzy godziny może się dobrze sprawdzić. Celem warsztatów jest wspólne dochodzenie do rozwiązania, a nie samo przedstawienie go.

13 Mecz

Mecz informatyczny (lub matematyczny) to rodzaj zawodów, w którym uczestnicy są podzieleni na dwie grupy i wśród tych grup wymyślają wspólnie rozwiązania zadań z listy przygotowanej przez kadrę, po czym tłumaczą sobie nawzajem rozwiązania. W drugiej fazie meczu wskazane osoby przedstawiają rozwiązania tych zadań aby zdobyć punkty dla swojej drużyny.

Szczecin ma swoją własną odmianę meczu informatycznego, którego starannie przygotowane zasady znajdują się [gdzieś tutaj](#).

Zasady zostały dobrane w taki sposób, aby uczestnicy starali się rozwiązać jak najwięcej zadań, a szczególnie, aby każdy z grupy nauczył się rozwiązań tych zadań. Dzięki temu osoby o większym poziomie będą zmuszeni wymyślać i tłumaczyć innym rozwiązania, a osoby o niższym – zrozumieć i zapamiętać rozwiązania.

Jest to znakomity sposób wymuszenia nauki wśród uczestników. Jednak należy mieć na uwadze to, że zasady meczu są tak brutalne, że część uczestników może się zestresować podczas prezentacji rozwiązań. W miarę możliwości należy być łagodnym i nie torturować zawodników tym meczem, ale nie można pokazać uczestnikom swojej łagodności, gdyż to obawa przed tymi torturami motywuje uczestników i wymusza naukę. Jest to też męcząca forma zawodów (przez stres i długą sesję wymyślania rozwiązań), dlatego najsensowniej jest, aby rozgrywka odbyła się ostatniego dnia.

Faza rozwiązywania zadań powinna być jak najdłuższa (czyli nie krótsza, niż jeden dzień). Z tego wynika, że jest jedna noc, w trakcie której uczestnicy cały czas starają się rozwiązywać zadania. Można wtedy zamówić pizzę dla uczestników (zależnie od budżetu obozu, albo na koszt obozu, albo na koszt uczestników), gdyż prawdopodobnie w nocy uczestnicy zgłodnieją, a lepiej do tego nie dopuścić podczas męczącego wieczoru (jest spora szansa, że uczestnicy sami by próbowali zamówić coś do jedzenia, więc gdy zapewni im się pizzę, to będą mogli bardziej się skupić na zadaniach). Trzeba przy tym uważać, żeby nie zabrać się za to za późno i zrobić dokładny rekonesans w okolicy ośrodka, bo informacje dostępne na Google Maps potrafią

być zwodnicze. Pizzerie bywają już zamknięte o północy.

14 Drużynówka

Drużynówka to zupełnie przeciwieństwo meczu. Są to zawody zazwyczaj w trzyosobowych drużynach z wykorzystaniem jednego komputera na zespół, w którym uczestnicy wspólnie rozwiązują i implementują większą ilość zadań, które często są luźniejsze lub zabawne. Jest to dobra opcja dla dywizji div3 i div4, ale niepotrzebna i nieskuteczna dla div1 i div2. Często drużynówka to dobra zabawa dla uczestników, ale trzeba zadbać o dostatecznie niski poziom trudności i różnorodność zadań. Część zadań może być niezwiązana z algorytmiką (mogą być zagadkami matematycznymi, albo czymś zupełnie innym, jak policzenie ile jest ławek w ośrodku), ale większość powinna jednak czegoś nauczyć zawodników (więc po drużynówce też jest potrzebne omówienie zadań).

15 Contest stały

Contest stały (contest dostępny przez cały obóz z pełnym raportem zadań i zapewnionym rankingiem, w którym uczestnicy dobijają zadania) to kluczowa część obozu. Uczestnicy najczęściej się uczą podczas implementowania / myślenia nad zadaniami, którymi zostali pokonani podczas zawodów. Należy więc się postarać, aby contest stały pojawiał się od razu po zawodach, aby było dostępne odpowiednie środowisko do dobijania zadań wieczorami / nocami oraz aby zawodnicy nie bali się pytać kadry zadaniowej o pomoc z przypomnieniem sobie rozwiązania, lub o pomoc z implementacją zadania. Dodatkowo można regularnie przypominać i motywować zawodników do dobijania contestu stałego (często zawodnicy są do tego niechętni, nawet jeżeli jest to najbardziej edukacyjna część obozu).

16 Zakończenie obozu

Nagrody dla poszczególnych dywizji są mile widziane. Można przyznać jakieś słodycze, owoce, maskotki lub wybrane zabawne nagrody dla kilku najlepszych osób z każdej dywizji. Sprawia to, że uczestnicy pozytywniej zapamiętują obóz oraz czują się dowartościowani po brutalnych contestach i meczach.

Sposób wybrania najlepszych uczestników powinien polegać na rankingu ogólnym lub na rankingu stałym (zależnie od dywizji), a najlepiej – na obu rankingach (można na przykład wybierać osoby na podstawie sumarycznej liczbie punktów zdobytych na contestach oraz na conteste stałym (z contestów i z zadań powykładowych). Można na samym początku obozu wyjaśnić, w jaki sposób będą ci uczestnicy nagradzani, aby zachęcić do dobijania zadań i rozwiązywania zadań powykładowych (przypominam, że zazwyczaj uczestnicy najczęściej się uczą dobijając zadania, a nie rozwiązując je na contestach).

Alternatywnie można nagradzać uczestników po każdym conteste za ich wyniki podczas tego contestu, a na zakończeniu obozu nagrodzić uczestników tylko za ranking stały.

17 Po obozie

Poza zmergowaniem SIM'a obozowego do SIM'a głównego (co należy zrobić jak najszybciej, aby osoby nie straciły ochoty na dobijanie zadań przez długie czekanie), nie ma niczego do zrobienia po obozie. Każdy jest już wtedy zmęczony (zarówno kadra, jak i uczestnicy) i nikt się nie pali do jakichkolwiek dalszych prac.

Dlatego jakiegokolwiek zapisywanie swoich błędów podczas prowadzenia obozu lub zapisywanie sugestii na przyszłe obozy powinno odbywać się podczas trwania obozu.

18 Recykling

Recykling zadań to czynność polegająca na ograniczeniu dostępu do zadań z obozu i niewykorzystywaniu tych zadań przez następne kilka lat. Dzięki temu, po dostatecznie wielu latach, można ponownie wykorzystać wszystkie zadania z tego obozu (albo złączyć lepsze zadania z drugim zrecyklingowanym obozem, aby powstał jeden, lepszy obóz), ponieważ żaden uczestnik nie zna tych zadań. Jest to technika warta rozważenia, jeżeli przypuszcza się, że stworzy się kilka obozów (a nawet, jeżeli nie, to jeżeli będzie to wykonywane w każdym roku przez organizatorów obozu, to i tak się to uda). Dzięki temu można liczyć na znacząco mniej pracy w przyszłości, co zwiększa czas na relaks.