

Propozycje zagadnień na zajęcia przygotowujące do Olimpiady Informatycznej

MOKIP

10 sierpnia 2010

Nie obijaj się!

Streszczenie

Poniższy wykaz ma na celu pomoc w doborze materiału przerabianego przed OI. Trzeba jednak zaznaczyć, że kluczem do sukcesu w tym konkursie jest zdecydowanie rozwiązywanie zadań. I to szczególnie zadań z poprzednich edycji OI - one powinny być przerabiane w ilościach „hurto- wych“. Osobiście przypadły mi do gustu także zadania z Potyczek Algorytmicznych, oraz Inżynierii Algorytmicznej (do znalezienia na stronach MIMUW). A w ogóle to im więcej zadań tym większe szanse. Podstawy teoretyczne są czasem pomocne, a czasem wręcz nieocenione, ale więcej pożytku przyniesie zrobienie paru zadań niż szczegółowe czytanie o kopcach dwumiennych.

Używane skróty myślowe to oczywiście Cormen - „Wprowadzenie do Algorytmów”, Lipski - „Kombinatoryka dla programistów“, BDR - Banachowski, Diks, Rytter - „Algorytmy i Struktury Danych”. Delfin - to nazwa pewnego obozu treningowego pod OI.

Kilka rzeczy jest używanych szczególnie często (wnioskuję to po częstości pojawiania się tych tematów na zajęciach, gdy omawiali mądrzejsi ludzie - ale mogę się mylić) i warto znać je jak własną kieszeń: DFS/BFS wraz z listową implementacją grafu, Przeszukiwanie Binarne, Dijkstra, Stos, Lista, Kopiec, struktura Find-Union, KMP, iloczyn wektorowy (choćby do sortowania kątownego, czy sprawdzenia czy para odcinków się przecina), statyczne drzewa BST. Trochę rzadsze, ale także warte są: Floyd-Warshall, Bellman-Ford, Sito Erastotenesa, alg. Euklidesa, sortowanie topologiczne, różne typy spójnych, cykle Eulera, otoczka wypukła i zawieranie się w wielokącie. Jeszcze rzadziej: Minimalne drzewa rozpinające, cykle Hamiltona, skojarzenie w grafie dwudzielnym, kombinacje liniowe („Dijkstra po resztach”), operacje na dużych liczbach... Ale tak naprawdę to wszystko co się przerzerobi pozostawia coś w głowie. Nawet jeżeli nie przyda się wprost w zadaniu to nauczy pomyślniku - tak więc materiał jest wart polecenia. Trzeba tylko pamiętać - prawdziwej wartości nabiera on dopiero po sporej liczbie odpowiednich, w pełni rozwiązanych zadań olimpijskich.

1 Wstęp - Podstawowe struktury i metody

Główne źródła: Cormen, BDR. Uwaga do prowadzącego: na tym etapie tylko przybliżać metody programowania (+ jakieś proste przykłady) - dogłębnie zostaną omówione w późniejszych działach.

- wprowadzenie: struktura danych jako zbiór z pewnym porządkiem oraz możliwymi do wykonania na nim operacjami.
- przykład: podzbiór \mathbb{N} , operacje: dodaj x , usuń x , znajdź x (sprawdź czy x znajduje się w zbiorze), zwróć x pierwsze (ostatnie) w porządku.
- notacja $O(n)$ (ponieważ chcemy aby operacje na zbiorze wykonywane były szybko potrzebujemy móc szacować czas wykonania). Ile operacji elementarnych przypada na jedną operację na zbiorze, w zależności od jego rozmiaru - przykład: implementacja na nieposortowanej tablicy.
- szacowanie oczekiwanej złożoności z rozmiaru danych
- korzystanie z porządku - tablica posortowana: Przeszukiwanie Binarne (jeden z najważniejszych algorytmów!)
- a gdy porządkiem ma być czas dodania do zbioru? Stos i Lista - implementacja wskaźnikowa. (element konieczny - nie iść dalej dopóki nie będzie to dla nich oczywiste!)
- kolejka priorytetowa - Kopiec (jeszcze ciągle wskaźnikowo, chyba łatwiej będzie im za pierwszym razem wyobrażać sobie to jako kółka i strzałki, jak już załapią to pokazać na tablicy)
- o wyższości struktur statycznych nad dynamicznymi: implementacja stosu, listy i kolejki na tablicy. (i w tej postaci będą zazwyczaj tego używać do zadań)
- „dziel i zwyciężaj“ - jeżeli potrafimy rozwiązać problem korzystając z rozwiązań jego mniejszych podproblemów
- rekurencja i iteracja - obliczanie silni i potęgi
- rekurencja + spamiętywanie = Programowanie Dynamiczne (gdy się da zastąpić rekurencję iteracją)
- przypomnieć: jeśli się da to używać struktur statycznych oraz iterować zamiast używać rekurencji - szybsze działanie i mniejsze zużycie pamięci.
- zadanie: odpowiadać na dużą liczbę zapytań o $\frac{a!}{b!} \pmod{c}$, $a \geq b$, $c = \text{const.}$ (przy tablicy - jakby jakimś cudem się zacięli to próbować prostrze wersje $a! \pmod{10}$, $a! \pmod{c}$, $\frac{a!}{b!}$)
- Programowanie Zachłanne (uwaga: robić dowody, bo można się przejechać)

2 Porządkowanie zbiorów

Główne źródła: Cormen, BDR.

- sortowanie tablic: przez selekcję, wstawianie, bąbelkowe, szybkie. (o złożonościach)
- zadanie na kilka minut: korzystając z poznanych struktur danych zrobić to w $O(n \log n)$; sortowanie przez kopcowanie, scalanie. O tym że porównując nie da się szybciej niż $O(n \log n)$.
- sortowanie liniowe: zliczanie, pozycyjne, kubełkowe. (w sporej liczbie zadań musimy posortować 10^6 liczb z zakresu do 10^6 - np wierzchołków w grafie - sortujemy je wtedy przez zliczanie).
- zadanie: Lolobrygida (VII-OI) ; omówić problem lidera.
- zadanie: Cormen 9.3-9 ; algorytm „magicznych piątek” - szukanie mediany i statystyk pozycyjnych (wraz z dowodem złożoności!)
- dlaczego nie wystarczy posortować tablicę by znać k -tą statystykę pozycyjną?
- I po co nam był ten dział? `std::sort()` . Jak używać sorta z własną funkcją porównującą.
- zamiast sortować duże obiekty możemy posortować tablicę ich indeksów - będzie szybciej. (wspomnieć o haszowaniu - lekkie podobieństwo)
- zadanie: Trawnik (Delfin styczeń 2008).
- Sztuczka z binary searchem po wyniku

3 Grafy I

Główne źródła: Cormen, Lipski, BDR. Na razie tylko grafy nieważone.

- reprezentacja grafu w komputerze: plik `*.bmp` ,macierz incydencji, sąsiedztwa, listy sąsiedztwa.
- przeszukiwanie w głąb, wszerz. O odległości wierzchołka od źródła. Jak potem wypisać ścieżkę od wierzchołka do źródła?
- spójne składowe
- skierowany graf acykliczny (DAG) i sortowanie topologiczne
- silnie spójne składowe i reprezentujący je DAG. zadania: Cormen 22.5-3, Drogi (Potyczki Algorytmiczne 2005)
- dwudzielność grafu i dwukolorowanie
- dwuspójne składowe, mosty, punkty artykulacji.
- Co się zmienia dla różnych typów spójnych składowych gdy graf jest skierowany/nieskierowany? Kiedy wogóle można o tym mówić?

- zadanie domowe: Biura (XIV-OI) ; omówić potem łącznie z implementacją.
- cykle Eulera i Hamiltona (odpowiednio też ścieżki) - warunki istnienia, sposoby znajdowania itp.
- fundamentalny zbiór cykli - znajdowanie, tworzenie pozostałych możliwych cykli w grafie.
- * 5-kolorowanie i planarność, graf dualny, o kolorowaniu państw na mapach.
- zadania: Billboardy (MWPZ-2006), Mrówki i Biedronka (VIII-OI), Blokada (XV-OI), Powódź (XIV-OI), Kości (XII-OI), Spokojna Komisja (OI-VIII), Linie Autobusowe (Potyczki Algorytmiczne 2005)
- * zadanie Narciarze (VII-OI) ; przekroje grafu, minimalny, maksymalny
- zadanie domowe: Dziuple (XII-OI); omówić potem grafy pladzielne

4 Struktury Danych

- problem: mamy zabawki w pudełkach. Na początku każda zabawka jest w swoim pudełku. Wykonujemy dużo operacji przesypania zawartości pudełka obecnie zawierającego zabawkę a do pudełka obecnie zawierającego zabawkę b . Chcemy w różnych momentach móc szybko odpowiadać na pytanie - jaki jest najmniejszy numer zabawki w pudełku w którym obecnie znajduje się zabawka c . Przedyskutować propozycje.
- Struktura zbiorów rozłącznych i jej banalna implementacja. (sławne Find-Union)
- Drzewa binarne - porządki pre/in/post-order.
- Przechodzenie wgląd z jawnym stosem - zadanie Chojnka z ASD2005-2 na SIO
- zadanie: zbudowano n -wierzchołkowe drzewo BST i wypisano je w kolejności:
 - preorder
 - inorder
 - postorder

Dany jest ów zapis. Stwórz je z powrotem i wypisz w kolejności

- preorder
- inorder
- postorder

Które z możliwych kombinacji są rozwiązywalne?

- n -wierzchołkowe drzewo wypisano w kolejności preorder oraz inorder. Mając te dwa zapisy odtwórz drzewo. Czy to jest wykonalne?

- * dwa n -wierzchołkowe drzewa - sprawdzić czy są izomorficzne. Uwaga! Drzewo = drzewo ukorzenione! (nawiązać do problemu sortowania drzew; zaznaczyć, że podobnie możemy porównywać i sortować inne szczególne typy grafów jakie się pojawią w zadaniu)
- problem najniższego wspólnego przodka
- omówić wzbogacane drzewa przeszukiwań binarnych: maksimów, licznikowe, pozycyjne, pokryciowe.
- *równoważenie drzew BST (przybliżyć sposób działania drzew czerwono-czarnych oraz AVL, drzewa samoorganizujące się - "splay")
- implementacja wzbogaczanych drzew BST na tablicy statycznej, przewagi nad `std::(multi)set/map` . (także wytłumaczyć jak korzystać z tych ostatnich).
- znajdowanie: k -ta statystyka pozycyjna (niech sami spróbują wymyślić przy tablicy).
- tablica sum prefiksowych zamiast drzewa (Jarowe czary na bitach - nie wiem jak to się wszystko nazywa)
- drzewa kartezyjańskie, budowa w czasie $O(n)$, jakiś prosty przykład zastosowania. (chyba można wykorzystać Paliwo z IV OI, ale pokazać też rozwiązanie oficjalne)
- drzepce (drzewce...)
- * złączalne kolejki priorytetowe (kopce dwumienne, Fibonacziego), można opowiedzieć
- zadania: Sortowanie (Delfin styczeń 2008), Małpki (X-OI), Klocki (XV-OI), Drzewa (XIV-OI), Komiwojażer Bajtazar (XI-OI), Listonosze (IA-MIMUW-2008)

5 Programowanie Dynamiczne i Zachłanne

Główne źródła: Cormen... i oczywiście Niebieskie Książeczki.

- linie montażowe (o własności optymalnej podstruktury)
- Standardy: mnożenia macierzy, optymalna triangulacja, plecak
- odtwarzanie optymalnego rozwiązania (ze spamiętanych wyborów częściowych)
- Przykłady: k -ta najkrótsza ścieżka w małym grafie bez wag; zadanie The Great Wall (ważne) z CERC 2004; ilość pokryć paska 3x kostkami domina; obliczanie $(n \text{ po } k)$; poszukać wśród zadań z OI - będzie pełno
- problem wydawania reszty
- zadanie domowe: Banknoty (XII-OI), Misie (XIII-OI)

- programowanie dynamiczne na drzewach: (schemat - z wyników dla synów obliczany wynik dla ojca), Zadanie: Liczność największego skojarzenia w drzewie, ile jest takich najliczniejszych skojarzeń (wyniki modulo k , w obliczeniach cząstkowych zastosować tablice iloczynów prefiksowych i sufiksowych - dobrze wyjaśnić dlaczego)
- zadania z przepinaniem korzenia: Stacja (XV-OI), Kontrolerzy (z konkursu MOKIP 2007/2008).
- problemy do rozwiązania: Cormen 15.5, 15-1,...,15-7
- oba problemy wyboru zajęć (ilość zajęć; łączny czas)
- zadanie domowe: Trójramienny Dźwig (VII-OI) ; z dowodem poprawności oczywiście
- zadanie: n czynności. Czas wykonania i -tej to $a_i + b_i \cdot l_0$, gdzie l_0 to czas jej rozpoczęcia. Dobrać kolejność tak, żeby w sumie zajęło jak najmniej.
- problemy do rozwiązania: Cormen 16.3, 16.5, 16-1,...,16-4
- dany jest zbiór stopni wierzchołków pewnego grafu. Zbudować dowolny graf nieskierowany w tym zbiorze stopni wierzchołków. (co zmienia jeśli to jest albo nie multigraf?)
- * matroidy (macierzowe, grafowe...) i ich algorytmy zachłanne
- koniecznie: BARDZO DUŻO ZADAŃ!

6 Kombinatoryka

Główne źródło: Lipski (rozdział 1). Można uzupełniać o różne rzeczy, np. jakieś "matematyki konkretne/dyskretnie", metody rozwiązywania rekurencji i inne. Ważne by zmuszać ich do myślenia, aby sami wyprowadzali i udowadniali różne rzeczy. Przecież na OI nie będzie raczej tych typowych zależności które zostaną omówione. Ale będą „Liczby Bajtazara” i będzie trzeba samemu znaleźć i udowodnić na nich jakieś prawa. (na przykład zależność rekurencyjną w celu konstrukcji programu dynamicznego).

- permutacje - rozkład na cykle, zliczanie inwersji, znak permutacji. (zadanie na 5 min: szyfrowanie korespondencji za pomocą permutowania jej 10^8 razy - okres permutacji). Lipski: zadanie 1.2.
- generatory permutacji: (anty) leksykograficzny, przez minimalną ilość transpozycji, przez minimalną ilość sąsiednich transpozycji. (wytłumaczyć zalety każdego z nich).
- zadania typu: dany jest zbiór jakichś obiektów kombinatorycznych T + jakiś porządek na nich. "Wyznaczyć k -ty w kolejności" lub "Który w kolejności jest x ".
Przykład: wyznaczyć pierwszą leksykograficznie n -permutację o dokładnie k inwersjach (również zadanie KOD z jednej z OI)

- rekurencje (zmuszać ich przy tablicy do prób rozwiązania): Wieże z Hanoi, Problem Flawiusza (razem z implementacją na bitach), Trójmian (XI-OI), dwumian Newtona (generowanie, dużo różnych zależności w Lipskim (poudowadniajcie sami przy tablicy); pamiętać: rekurencja+spamiętywanie=dynamik - tak dla przypomnienia słuchaczom)
- generatory: wszystkich podzbiorów zbioru (binarny kod Greya!), podzbiorów z powtórzeniami, podzbiorów k -elementowych.
- szeregi: geometryczny, arytmetyczny, harmoniczny (także wersie an-), ich sumy, $1^2 + 2^2 + \dots + n^2$, $1^3 + 2^3 + \dots + n^3$, jak znaleźć $1^k + 2^k + \dots + n^k$? (konieczne zrozumienie skąd się bierze każdy wzór - daje to większy pożytek niż bierne zapamiętanie)
- NA BLACHEŃ WZORY:
 - suma szeregu arytmetycznego
 - $1 + 3 + 5 + \dots + (2n - 1)$
 - $1^2 + 2^2 + 3^2 + \dots + n^2$
 - $\binom{n}{k} = \frac{n!}{k!(n-k)!} = \binom{n-1}{k} + \binom{n-1}{k-1}$
- zadanie domowe: KOD (VII-OI); potem omówić Liczby Catalana
- generacja podziałów zbioru
- liczby Styrlinga II rodzaju, Liczby Bella (zależności wyprowadzać na tablicy - muszą umieć znajdować takie wzory)
- generacja podziałów liczby w porządku antyleksykograficznym
- zasada włączania/wyłączania. Zbiory i prawdopodobieństwa. Małe zadanko domowe: zbiór n różnych liczb, losujemy ciąg k elementów (bez powtórzeń) - jakie jest prawdopodobieństwo aby ciąg był ściśle monotoniczny?
- * Liczby Styrlinga I rodzaju i ich związek z wielomianami
- równania rekurencyjne, nawet te trudne i brzydkie (równania funkcyjne też można do tego dołożyć)
- * Funkcje Tworzące

7 Grafy II

Główne źródła: Cormen, Lipski, BDR. Grafy ważone.

- oczywiście wszyscy na sali wiedzą jak to zaimplementować?
- problem najkrótszych ścieżek: (między wybraną parą jest równoważny wszystkim parom z jednym źródłem) relaksacja i algorytm Bellmana-Forda. (dynamiczny). A gdy jest cykl ujemny?

- przypadek dla wag nieujemnych: algorytm Dijkstry (zachłanny) - znaleźć kontrprzykład z wagami ujemnymi. (pokazać pewne podobieństwo do BFS). Omówić złożoność i implementację. (jeden z najczęściej stosowanych algorytmów)
- sposób zamiany wag ujemnych na nieujemne z zachowaniem porządku odległości (aby można było użyć Dijkstry).
- odległości między wszystkimi parami wierzchołków: algorytm Floyda-Warshalla (dynamiczny)
- * opowiedzieć o algorytmie Johnsona dla grafów rzadkich, kiedy chcielibyśmy go stosować a kiedy nie?
- Najkrótsze ścieżki w DAG (niech spróbują wymyślić przy tablicy)
- Minimalne Drzewo Rozpinające: algorytmy Kurskala i Prima (dobre przypomnienie - niech sami spróbują wpisać czego użyć czego użyć do scalania drzew w Kurskalu).
- zadanie domowe: Zwiedzanie Miasta (VIII-OI) ; potem na zajęciach omówić problem stacji paliwowych na cyklu (ładnie się to pokazuje rysując i przepinając, z początku trasy na koniec, strzałki reprezentujące zmiany stanu baku)
- przepływy w sieciach (twierdzenie o maksymalnym przepływie, ścieżkach rozszerzających i minimalnym przekroju), przegląd prostrzych metod - Ford-Fulkerson, Dinic, Edmonds-Karp (bez jakiegoś tam terroru - wystarczy że wiadomo że wystarczy puszczać rozszerzającego DFSa jak długo się da i tyle)
- * maksymalny przepływ dla krawędzi jednostkowych
- * najtańszy/najdroższy maksymalny przepływ dla krawędzi jednostkowych, Algorytm Węgierski
- najliczniejsze skojarzenie w grafie dwudzielnym (wersje $O(n \cdot (n + m))$) oraz Hopcroft-Karp $O((n + m) \cdot \sqrt{n})$, * dowód złożoności ostatniego)
- zadania typu: po osi liczbowej skacze pchła, może wykonywać następujące skoki:
 - jeśli jest na pozycji podzielnej przez 21 to może się przemieścić o 6 do przodu lub 21 do tyłu
 - tutaj 51 takich podobnych warunków
 - do tego z każdym ruchem wiążemy koszt

Jakim minimalnym kosztem możemy dostać się z zera na pozycję n ?

- dana jest tablica $t[a, b]$ gdzie $t[a, b]$ to koszt kupna waluty b za walutę a otrzymaną w serii wymian zgodnie z cennikiem t
Wykryć sytuację, gdy jakąś walutę można pomnażać w nieskończoność.
- n -ta najkrótsza ścieżka dla małego n ; ścieżka z a do b o długości dokładnie k , zadanie: Zamek (IX-OI)

- zadanie domowe: Przemysłowicy (X-OI), Skoczki (OI krajów Bałtyckich (BOI) 2001)
- najdłuższe ścieżki (w drzewach ważonych, w grafach - problem do dyskusji)
- Szybkie podnoszenie grafu do potęgi (binarne)
- zadania: Podróż (potyczki 2008) - omówić potem na zajęciach problem ilości ścieżek między parą wierzchołków w grafie nieważonym. A co jeśli długość ścieżki jest ograniczona, a co jeśli graf jest ważony?

8 Teoria Liczb

Główne źródło: Cormen. Można dodać jakieś mądre książki z matematyki.

- działania w ciele \mathbb{Z}_p (czyli modulo p , p - liczba pierwsza)
- grupy \mathbb{Z}_n^* , dlaczego $\phi(n) = |\mathbb{Z}_n^*| = \prod_{p|n} (1 - \frac{1}{p})$, gdzie $p \in \mathbb{P}$ - własność przyda się później.
- zwrócić uwagę na ZAKRES ZMIENNYCH przy mnożeniu np. $10^5 \cdot 10^5 \pmod{200000}$ będzie źle policzone na intach.
- jednoznaczność rozkładu liczby na czynniki pierwsze (jeśli jakimś cudem komuś by o tym nie powiedzieli w podstawówce/gimnazjum)
- liczby względnie pierwsze, NWD i NWW, algorytm Euklidesa (omów złożoność - liczba n ma pesymistycznie $\log n$ dzielników ; czasem szybciej dzielić przez 2 - „binarny algorytm Euklidesa“)
- NWD jako najmniejsza dodatnia kombinacja liniowa, rozszerzony algorytm Euklidesa
- Pogrupy generowane przez jeden element i cykliczność operacji: grupa (S, \oplus) , $\text{ord}(a) | |S| \Rightarrow a^{|S|} = e$ (tw. przyda się potem), przykład: grupy potęg $a^t \pmod{n}$ - grupa generowana przez liczbę a i operację mnożenia w grupie modulo n .
- szybkie potęgowanie modularne: a) przez znalezienie okresu , b) przez szybkie potęgowanie a^b w $O(\log(b))$ (to gdzie co krok mnożnik podnosimy do kwadratu).
- liczba odwrotna do danej:

$$n^{-1} \equiv x \pmod{p}$$

w.t.w (z definicji)

$$x \cdot n \equiv 1 \pmod{p}$$

obliczanie x przy pomocy ROSZERZONEGO ALGORYTMU EUKLIDESA

- rozwiązywanie modularnych równań liniowych $ax \equiv b \pmod{n}$

- chińskie twierdzenie o resztach
- sita liczb pierwszych i tablice faktoryzacji: Sito Erastotenesa $O(n \log n)$, $O(n \log \log n)$ - to drugie najszybsze do zadań, a tylko jeden znak w kodzie różnicy od zwykłego, *dowód złożoności tego drugiego. $O(n)$ - omówić na rysunkach struktur dynamicznych, potem przedstawić implementacje na dwóch statycznych tablicach. Ilość liczb pierwszych $\lim_{n \rightarrow \infty} \frac{\pi(n)}{\frac{n}{\ln n}} = 1$.
- testy pierwszości: deterministyczny $O(\sqrt{n})$. Tw. Eulera ($a^{\phi(n)} \equiv 1 \pmod{n}$), Małe Tw. Fermata $p \in \mathbb{P} \Rightarrow a^{p-1} \equiv 1 \pmod{p}$, Liczby Carmichaela - opowiedzieć co to i jak często się pojawiają; tw. $p \in \mathbb{P} \setminus \{2\} \wedge x^2 \equiv 1 \pmod{p^n} \Rightarrow |x| = 1$. Test Millera-Rabina (heureux, ale działa dla podstaw $\{2,3,5,7\}$ w zakresie do 2000000 (000?)). Istnieje test deterministyczny w $O(\log^{12} n)$, a podobno w $O(\log^8 n)$.
- obliczanie odwrotności BEZ rozszerzonego euklidesa, w \mathbb{Z}_p (z małego fermata)! $a^{-1} \equiv_p a^{p-2}$
- zadanie: dwa wiadra o pojemnościach v_1 i v_2 ; zbiornik pusty o dużej pojemności wraz z nieskończonym źródłem wody. Nalać do zbiornika dokładnie n litrów.
- kombinacje liniowe: o dodatnich współczynnikach - „Dijkstra po resztach”; o współczynnikach całkowitych - NWD.
- * heurystyczny rozkład na czynniki ρ Pollarda (Jaro: ”Olać biedne ro pollarda, nie jest zbyt interesujące”)
- zadanie domowe: Cormen - problem 31-4, Wyliczanka (IX-OI), Zapytania (XIV-OI), Sumy Fibonacciego (XII-OI); omówić potem: własności liczb Fibonacciego, (ważne Cormen - problem 31-3). Systemy pozycyjne i ich zamienianie ze sobą (np z trzynastkowego na sześćdziesięć siódmkowy i na odwrot), także system Zeckendorfa i inne podobne cudowne systemy
- Wzór Stirlinga $\lim_{n \rightarrow \infty} \frac{n!}{\sqrt{2\pi n} (\frac{n}{e})^n} = 1$
- * metoda sita kwadratowego
- * problem logarytmu dyskretnego

9 Algorytmy Tekstowe

Główne źródła: Cormen, BDR.

- problem wyszukiwania wzorca: Algorytm Naiwny
- Schemat Hornera i algorytm Rabina-Karpa, haszowanie (można użyć kilku niezależnych tablic haszujących). Dlaczego i kiedy ten algorytm jest heurystycznym? (lub uzupełniony o sprawdzanie naiwne jest pesymistycznie kwadratowy)

- Funkcja Prefiksowa i jej generacja (bardzo ważny moment - muszą absolutnie to rozumieć)
- wyszukiwanie wzorca KMP (z dowodem złożoności liniowej)
- drzewa prefiksowe, kompresja drzewa podslów
- przypadek wielu wzorców - Algorytm Aho-Corasica, przypadek dwuwymiarowy - algorytm Bakera.
- przypadki zdegenerowane: symbole uniwersalne (jedno, kilku i wiele znakowe).
- *można opowiedzieć o wyszukiwaniu wzorca metodą automatów skończonych
- dany jest zbiór szablonów i słowo. Na ile sposobów można otrzymać słowo przez konaktencje szablonów (mogą się powtarzać), np: {r,ry,yt,ter,t} rytter? Odp. 2 bo: r-ry-t-ter , r-r-yt-ter
- *jak to zrobić w czasie liniowym względem rozmiaru wejścia + ilości wystąpień szablonów w słowie
- czy x jest cyklicznym przesunięciem y ? $O(|x|)$
- najdłuższe wspólne pod słowo $O(???)$
- najdłuższy wspólny podciąg $O(N \cdot M)$
- minimalny okres słowa, maksymalnie leksykograficzny sufix - algorytm Dinica, minimalna leksykograficznie cykliczność słowa.
- (*) algorytm KMR
- *tablice sufiksowe i przykłady zastosowań
- *przybliżyć drzewa sufiksowe i grafy podslów
- zadanie domowe: Palindromy (XIII-OI); potem omówić wszelkie metody (dobre i złe).
- algorytm Manachera (pokazać że palindrom ma bardzo zorganizowaną budowę; Uwaga: całkiem spro zadań z palindromami nie korzysta wcale z tego algorytmu - żeby nie próbować go na siłę szukać w zadaniach)
- zadanie domowe: Wirusy (VII-OI), Szablon (XII-OI), MegaCube (V ObózOI im. A. Kreczmara), Palindromy (II-OI)

10 Geometria Obliczeniowa

Główne źródła: Cormen, BDR.

- czy punkt znajduje się po lewej czy po prawej stronie wektora? - iloczyn wektorowy (najważniejsze pojęcie - bez tego nie da się iść dalej); istotne jest by stosować operator `inline` (jeżeli tylko nasza funkcja nie „włazi” do STL)

- czy dwa odcinki się przecinają
- pole równoległoboku i trójkąta
- obliczanie pola wielokąta: z trójkątów; z trapezów (jeżeli wielokąt ma wierzchołki o współrzędnych całkowitych to jego podwojone pole jest liczbą całkowitą)
- ZAMIATANIE: po współrzędnych
- czy punkt należy do wielokąta?
- zapytać czy wiedzą jak implementować zbiory punktów z zakresu nie mieszczącego się w pamięci (np $|x|, |y| \leq 10^8$, ilość punktów poniżej 10^6), aby łatwo można było sprawdzić czy jakiś punkt się zawiera w zbiorze czy nie - powinni wiedzieć (jak nie to lanie).
- Sortowanie Kątowe
- ZAMIATANIE: po kątach
- otoczki wypukłe: algorytmy Grahama, Jarvisa, przyrostowy, dziel i zwyciężaj; problem otoczki wypukłej „on-line“
- ze zbioru punktów wybrać trójkąt o największym polu
- mając do dyspozycji n odcinków podać maksymalne pole wielokąta złożonego z nich wszystkich (dowód indukcyjny twierdzenia, że wielokąt ma największe pole, gdy możemy go wpisać w okrąg; jak znaleźć promień takiego okręgu i pole wielokąta)
- zadanie domowe: Przekątne (Potyczki Algorytmiczne 2008), Ołtarze (VI-OI)
- odległość punktu od prostej
- okręgi: przecinanie się, reprezentacja za pomocą trzech punktów, zadanie: Akcja Komandosów (XII-OI)
- zadania: Szachownica (Pot. Alg. 2008), Bilard (Pot. Alg. 2005), Szklana Pułapka (XV-OI); omówić potem różne zastosowania funkcji liniowej;
- para najbliższych punktów
- para najdalszych punktów
- czy kórakolwiek para odcinków się przecina? Wypisz wszystkie pary przecinających się odcinków.
- zadanie domowe: Osie Symetrii (XIV-OI)
- zadanie: Kopalnia Złota (VIII-OI)
- WAŻNE, TRUDNE: jak radzić sobie z przypadkami zdegenerowanymi:
 - odcinki pionowe i poziome
 - odcinki zdegenerowane

– wielokąty z wierzchołkami na środku boków

- ZADANIE NA KONTESTA: zbiór prostokątów; policzyć ilość rozłącznych obszarów nie pokrytych. Przykłady: (obie wersje dotyczące rogów na koncieście ≥ 3 godziny)
- metryki: nowojorska, rzeka, studnia, maksimum, wygląd okręgu w poszczególnych, zadanie Magazyn (XIII-OI), uniwersalny sposób na zamianę jednej metryki w drugą
- *inne geometrie (nieeuklidesowe, na sferze itp.)
- zadanie domowe: Punkty (XII-OI), Marsjańskie Mapy (OI krajów bałtyckich (BOI) 2001), Puste Prostopadłościany (VI-OI)

11 Problemy NP-zupełne i trudnorozwiązywalne

Główne źródła: Cormen, zadania

- zadanie atrakcje turystyczne (XIV-OI)
- problem komiwojażera
- Zadanie Zosia (XIII-OI)
- problemy klikli i pokrycia grafu
- problem pakowania i jego modyfikacje (np zadanie - Łódka - IA-MIMUW-2008)
- krótkie przybliżenie sporej liczby popularnych problemów NPC, w celu szybkiego ich rozpoznawania na zawodach. (wskazówka: czasem można rozpoznać takie zadanie przez dziwne ograniczenia rozmiaru danych, jedna zmienna bardzo duża - żeby brut nie przeszedł, a druga bardzo mała - żeby dało się zrobić jakiś algorytm dla tego szczególnego przypadku)
- problem izomorficzności grafu, *proponowany algorytm wielomianowy (Reiner Czerwinski)
- *problem sumy podzbiorów

12 Algebra

Główne źródła: Cormen

- eliminacja Gaussa
- eliminacja Gaussa w \mathbb{Z}_2 , w \mathbb{Z}_p
- ****algorytm sympleks
- zadania: Taniec (Szwajcarska OI 2004), Sejf (Potyczki Algorytmiczne 2006), *Szalony malarz (*acm.zju.edu.cn*)

13 Teoria Gier

Główne źródła: XI-OI Niebieska Książeczka

- zadanie: Sznurki (OPSS)
- stany gry - pola wygrywające i przegrywające
- gra NIM
- gra HEX
- gra w „Czekoladę”
- liczby NIM i łączenie gier
- ??? (zadania/przykłady?)

14 Duże liczby

Główne źródła ???

- implementacje dużych liczb naturalnych, całkowitych, wymiernych - zapis pozycyjny oraz przez rozkład na czynniki
- niech każdy zaimplementuje przed zawodami przynajmniej raz podstawowe operacje: +, -, ·, porównanie, wczytanie, wypisanie, potęgowanie binarne, dzielenie przez 2
- zadanie na 10 minut: policzyć największą liczbę Mersenne’a m mniejszą od $n < 10^{1000}$, k -ta liczba Mersenne’a: $2^k - 1$
Zrobić dyskusję na temat rozwiązań
- Policzyć na kontesta:

$$\sum_{i=1}^n \frac{2^{NWD(i,n)}}{n}$$

przedyskutować rozwiązania

- *udowodnić, że podany wzór określa ilość różnych kolorowań n -koralikowego naszyjnika (ale takiego, którego przód jest zawsze z jednej strony, czyli że nie można go odbijać symetrycznie. Dla $n = 4$ mamy 6 naszyjników)
- NWD na dużych liczbach (nie z algorytmu Euklidesa lecz ze wzoru, problem Cormen 31-1,2):

$$\gcd(\text{parzysta1}, \text{parzysta2}) = 2\gcd(\text{parzysta1}/2, \text{parzysta2}/2)$$

$$\gcd(\text{parzysta1}, \text{nieparzysta2}) = \gcd(\text{parzysta1}/2, \text{nieparzysta2})$$

$$\gcd(\text{nieparzysta1}, \text{nieparzysta2}) = \gcd(\text{nieparzysta1}, \text{nieparzysta1} - \text{nieparzysta2})$$

$$\text{gdy } \text{nieparzysta1} \geq \text{nieparzysta2}$$

- udowodnić że wykonywanych jest $O(\log^2 n)$ operacji przy liczeniu $NWD(a, b)$
- zadanie: Kafelki (X-OI)
- zadania domowe: Łańcuch (VIII-OI), Lampki (XV-OI)
- jak szybko możemy mnożyć duże liczby? a jak szybko dzielić? - dyskusja, kto szybciej.

15 Różności

- zadanie Kodowanie Permutacji (finał II OI)
- zadanie Autostrady (X-OI) - Ważne!
- zadanie Wędrowni Treserzy Pcheł (VIII-OI) - Ważne!
- zadanie Superskoczek (IX-OI)
- Układy SAT2 (czy coś takiego - nie wiedziałem do jakiego działu to dać) i układy logiczne wogóle
- technikalnia: optymalizacje, `inline`, operacje na bitach
- `bit_vector` ?
- zadanie Liczby Antypierwsze (VIII-OI) - o preprocessingu: kiedy stosować, jak go używać w implementacji
- kolejność komórek w tablicy wielowymiarowej
- bruty i heurezy (młotki: żółw, nóż) - na OI nie zostawia się zadań niewyślanych - każdy punkt może być decydujący (np. próg 101 pkt.). Zazwyczaj lepiej puścić optymalizowanego bruta niż heurystykę - trochę opowiedzieć o wyczuciu w pisaniu młotków. O młotkach przez haszowanie (np do Pociągi z XV-OI i inne).
- algorytmy randomizowane oraz funkcje „prawie losowe“ (np. $f(x) = ((x^2 - c) \bmod n)$).
- pisanie generatorów testów, brutowych sprawdzaczek i skryptu automatyzującego generację i porównywanie. Jest to dość ważne - często napisanie generatora, bruta i skryptu zajmuje stosunkowo niewiele czasu, a jak już się puści testy, to niech lecą w tle do końca zawodów (o ile koniecznie nam nie jest potrzebna do czegoś moc obliczeniowa). Praktycznie eliminuje to zjawisko głupich błędów. (Wiem z doświadczenia - gdybym nie puścił dużej ilości losowych testów na jednym zadaniu to nie znalazłbym durnego buga, i nie wszedłbym do finału!)
- *Automaty i gramatyki - koniecznie z jakimiś zadaniami. Pomocne mogą być zajęcia z Teorii Automatów w Pałacu Młodzieży (prowadzi je dr. Adam Kolany - można poprosić go o jakieś wskazówki o czym powinno się opowiedzieć), zadanie domowe: Genotypy (IV-OI)
- zadania, zadania, zadania...

I Etap			II Etap			III Etap		
OI	ZAD	100p	OI	ZAD	100p	OI	ZAD	100p
X	CZE	587	IX	IZO	226	IX	MIN	42
VIII	ANT	491	XIV	GRZ	148	XIV	ODW	38
XIII	KRA	474	IX	KUR	81	XII	AUT	27
XII	SKA	446	X	MAS	77	X	GRA	23
XI	PIN	429	XII	BAN	55	XII	DWU	22
VIII	MAP	397	XI	MOS	53	IV	KAJ	22
IX	WYS	331	XII	LOT	43	IV	REZ	20
VI	GRA	305	VI	BIT	36	VIII	LAN	20
XIII	OKR	303	XIV	MEG	34	IV	ALI	19
XII	SAM	271	VII	TRO	30	XIV	GAZ	19
XII	SKO	237	XII	SUM	28	V	BAN	16
V	SUM	230	VII	KOD	27	XIII	MIS	12
XII	BAN	219	IV	PAL	24	VII	POW	11
XI	SZP	208	IV	ADD	20	V	PRO	11
VIII	PRZ	202	IX	WYL	20	VI	WOD	10
IX	KOM	174	XIV	TET	19	XIV	WAG	10
XIV	OSI	167	VIII	GOR	19	VI	MAP	8
X	CIA	151	XIII	NAJ	17	X	MAL	8
XIV	BIU	144	VIII	POD	16	VII	POL	8
IX	ZAM	144	XI	PRZ	15	XII	DZI	7
X	PRZ	138	XIV	POW	14	IV	LIC	7
XIII	PRO	118	VIII	WYS	12	XIV	KLO	7
VI	MON	117	IX	PRO	10	IX	WAG	6
VII	NAR	114	V	PLE	9	XIII	EST	6
IV	TAN	110	X	POL	9	XI	MIS	6
X	LIC	109	XIII	MET	9	XI	ZGA	6
IX	KOL	109	X	KAF	7	XIV	KOL	6
XIV	ZAP	83	VII	LAB	6	XI	WYS	5
XI	ZAW	75	XIII	MAG	6	V	GRA	5
V	WIE	71	XIII	SZK	5	XII	AKC	4
IV	SKO	68	XIV	SKA	4	XIII	KRY	3
X	PLY	67	XIII	ORK	4	XI	WSC	3
VI	PUS	66	VI	LUN	4	XII	PRA	3
V	PRA	64	XII	SZA	4	IX	NAR	3
IX	SUP	61	XI	BRA	3	X	SUM	3
VII	GDZ	58	V	ROW	3	XIV	EGZ	3
IV	XOR	52	V	OKN	3	VIII	ZWI	3
VI	MUS	48	XII	KOS	3	VII	JAJ	2
XII	PUN	33	XIII	LIS	3	IV	TRO	2
XIV	ATR	32	IX	DZI	2	VI	PIE	2
XI	SZN	31	X	TRO	2	VII	PRO	2
XI	GRA	21	V	PAK	2	VI	MAG	2
VIII	GRA	20	VII	AUT	2	XI	KAG	2
XIII	ZAB	19	XI	TUR	2	XIII	PAL	2
XIII	TET	16	VI	GRO	1	VII	AGE	2
IV	LIC	16	X	AUT	1	X	TAS	2
VII	WIR	12	IV	REK	1	X	SKA	1
XIV	DRZ	5	VIII	MRO	0	XI	MAK	1
VII	PAS	1	VI	LOD	0	VI	OLT	1
V	ABS	0	VIII	SPO	0	XIII	ZOS	1
			IV	GEN	0	XII	LUS	1
			XI	JAS	0	XII	DWA	1
						IX	LIC	0
						IX	NAW	0
						V	GON	0
						IX	SZY	0
						XIII	TAN	0
						X	KRY	0
						V	NAJ	0
						V	LAM	0
						VIII	NAS	0
						VIII	BAN	0
						VIII	KOP	0

Przydatne Adresy v0.2

- *oi.edu.pl* - Oficjalna strona OI. Zawiera wszystkie zadania z olimpiad, oraz Niebieskie Książeczki. Te właśnie zadania powinny być robione przede wszystkim! (rozwiązania należy testować pod *main.edu.pl*)
- *sio.mimuw.edu.pl* - System internetowy OI. Służy do zgłaszania zadań. Zawiera zadania i linki do różnych zawodów.
- *main.edu.pl* - Zadania z Potyczek Algorytmicznych, Pogromców Algorytmów, spis konkursów, trochę samouczków. Możliwość zgłaszania rozwiązań z OI czy PA.
- *wazniak.mimuw.edu.pl* - Materiały Instytutu Matematyki Informatyki i Mechaniki. - Sporo przydatnych wykładów.
- *mimuw.edu.pl/~idziaszek/zajecia/ia/* Zadania z Inżynierii Algorytmicznej na MIMUW (jakaś sprawdzaczka, ale chyba nie ogólnodostępna do tego jest tu *dijkstra.niedasie.mimuw.edu.pl/user.phtml?c=4883*)
- *ki.staszic.waw.pl* - Kółko Informatyczne Staszica - zadania treningowe pod OI, trochę materiałów.
- *konkurs.adb.pl* - Strona Potyczek Algorytmicznych - konkurs, zadania.
- *mwpz.poznan.pl* - Strona konkursu drużynowego (w stylu ACM).
- *topcoder.com* - Strona konkursu TopCoder.
- *ioinformatics.org* - Strona Międzynarodowej OI. Zadania z międzynarodówek - warte polecenia.
- *opss.safo.biz* - Serwer zadaniowy, zawody, zawody stałe. (PL)
- *cm2prod.baylor.edu* - Strona ACM ICPC - zadania z tych zawodów.
- *acm.sgu.ru* - Serwis konkursowo zadaniowy.
- *acm.uva.es* - Serwis konkursowo zadaniowy.
- *acm.zju.edu.cn* - opis j.w.
- *spoj.sphere.pl* - Serwis konkursowo zadaniowy.
- *pl.spoj.pl* - „Polski SPOJ“ - opis j.w.
- *sgi.com/tech/stl* - Dokumentacja STL. Wszystko o sort, (multi) set/map i innych.
- *ann.matnness.net* - Przyjazny opis STL po Polsku. Trochę C++, algorytmów, struktur itp.
- *cplusplus.com* - Zawiera referencje funkcji C++. Nie wiesz dokładnie jak używać czegoś (np. scanf) - tu dokładnie pisze co jest co.
- *om.edu.pl* - Oficjalna strona OM. Zadania z wszystkich OM. Rozwiązywać je także powinni chętni do OI, ponieważ to uczy myśleć.

- *spinor.stolorz.pl* - Strona Pracowni Matematyki Pałacu Młodzieży w Katowicach. Informacje o konkursach oraz ZAJĘCIACH. Konkursy to (między innymi):
 - Rozkosze Łamania Głowy - klasy: I, II LO
 - Nudna Matematyka - klasy: IV - VI SP., I - III Gimn. oraz I szk. ponadgimnazjalnych
 - Ogólnopolski Sejmik Matematyków - klasy: I - III LO
- *matwbn.icm.edu.pl/ksspis.php* - Biblioteka Matematyczna (Sierpiński i wielu innych!!!)
- *skm.katowice.pl* - Śląski Konkurs Matematyczny (do II LO włącznie)
- *mat8lo.phg.pl/konkursy/default.php* - Spis konkursów matematycznych wg. Pikowej Pracowni Matematyki. Oficjalna strona tejże pracowni.
- *wm.staszic.waw.pl* - Strona Warsztatów Matematycznych wg. Staszica. Cel: trening pod OM.
- (to nie na oi) Kodi: "*mimuw.edu.pl/~krych/matematyka/ mimuw.edu.pl/~henrykm/analiza2007/* [ja:]ło cholera - ale zadania ma ten Henryk!" (tu są chyba zadanka ciekawe, miałem z tym kołosem 1 semestr i zdań jest w cholere dużo, ale sporo jest trudnych)"

Przydatne Książki

- „Niebieskie Książeczki”
- T. Cormen - „Wprowadzenie do Algorytmów” - W zasadzie absolutna podstawa i kręgosłup nauki algorytmów.
- W. Lipski - „Kombinatoryka dla Programistów” - Jak nazwa wskazuje. Szalenie przydatna.
- Banachowski, Diks, Rytter - „Algorytmy i Struktury Danych” - Uzupełnia poprzednie książki głównie o zbiór algorytmów tekstowych. Przydatna.
- „Matematyka Konkretna” i „Matematyka Dyskretna” - ludzie mówią że to dobre i przydatne książki - ja nie wiem, nie czytałem.
- E. Knuth „Sztuka Programowania” - w zasadzie taki sam opis jak wyżej.
- Cytując Accka: „Oprócz tego fajnym podręcznikiem jest seria ”Algorytmy w C++” Sedgewicka. Niestety chyba jeszcze nie cała została przetłumaczona na Polski, więc nie obejmuje całego materiału, ale jest według mnie napisana bardzo przystępnie.
Dla dość początkujących, chcących się dowiedzieć trochę bardziej, o co chodzi w algorytmice, można wykorzystać ”Algorytmy” M. Sysły.”

- zbiory zadań z Olimpiad Matematycznych Henryka Pawłowskiego, „Białe Książeczki“ , suplementy z kółka olimpijskiego w Katowicach - dlaczego umieszczam zbiory zadań pod OM w liście? - ponieważ matematyka najbardziej rozwija myślenie, a to się bardzo przydaje w OI (żeby nie powiedzieć - jest niezbędne). Uważam że warto przeplatać zadania między OI i OM. (dla młodszego rodzeństwa: zbiór zadań Dworeckiej (podstawówka i wczesne gimnazjum))
- książki z matematyki Sierpińskiego, (a także Lwa Kurlyandchika - dużo ładnych ”perełek” matematycznych)
- * C. Godsil, G. Royle ”Algebraic Graph Theory” (Jaro: ”Jak weźmie się algebre i wepchnie do teorii grafów to dostaje się tak CHOLERNIE silne narzędzie, że to jest niesamowite. Siła tego czegoś jest podobna jak analityczne rozważanie problemów teorii-liczbowych, przy czym dużo ładniejsze, i prostsze dla zwykłego śmiertelnika”)
- (to już nie na OI) Kodi: ”zadania z analizy matematycznej (3 tomy)Kaczor-Nowak - fajny zbiór [ja:”faktycznie zawiera zadania mocne - warto“], krysicki-włodarski (nie znam pełnej nazwy) ma proste zazwyczaj zadanka na rozruszanie się”

Zadania popularne

... czyli takie na temat których pojawiały się rozmowy. Może to świadczyć o nieco łatwiejszym poziomie trudności, może o zadaniu z pomysłem, może o przyjemnej formie zadania, rzadziej o zadaniu nieco trudniejszym. Umyślnie nie zamieszczałem ich wszystkich w blokach tematycznych, gdyż niekiedy zorientowanie się z jakiej dziedziny zadanie pochodzi już jest kluczem do rozwiązania.

Po co komu spis o tak dziwnym kryterium? Myślę, że można mieć czasem moment, że nie wiadomo od którego zadania zacząć. Może wtedy taka propozycja okaże się pomocna. Jest to też forma losowania sobie zadań. Na pewno nie zaszkodzi więc tego tu umieścić.

- IV-OI: Addon, Paliwo
- VI-OI: Lunatyk, Ołtarze, Pierwotek Abstrakcyjny, Trójkolorowe Drzewa Binarne, Woda
- VII-OI: Gdzie Zbudować Browar, Wirusy, Narciarze, cały II etap, Lolo-brygida, Agenci, Promocja
- VIII-OI: Liczby Antypierwsze, cały II i III etap
- X-OI: Przemysłowcy, cały II etap, Skarb, Małpki
- XII-OI: cały II etap, Dziuple, Dwuszereg, Dwa Przyjęcia, Akcja Komandosów, Prawoskrętny Wielbłąd, Autobus
- XIII-OI: Krążki, Profesor Szu, Szkoły, Magazyn, Orka, Palindromy, Misie

- XIV-OI: cały I etap, Grzbiety i Doliny, Powódź, Megalopolis, Tetris Attack, Egzamin Na Prawo Jazdy, Waga Czwórkowa
- XV-OI Szklana Pułapka, Robinson, Klocki, Mafia, BBB, Blokada, Lampki, Stacja
- Potyczki Algorytmiczne 2008: Przekątne, Sejf, Studia, i w ogóle wszystkie inne...

Ćwiczenia tematyczne

... mają na celu przećwiczenie przerobionego materiału. Podział wg. zewnętrznej pracy naukowej. Ćwiczenia te nie mogą zastąpić w części ni w całości zadań z OI, ale poćwiczyć zawsze warto, bo nawet jak coś się dobrze rozumie, a się tego nie zaimplementuje w domu, to na zawodach będziemy mieli problem (np. popełnimy błąd, albo zużyjemy za dużo czasu). Wprawa także ma znaczenie.

- Algorytmy Grafowe:
 - BFS:
 - * Łatwe: acm.uva.es: 10187, 10959. spoj.sphere.pl: 206. acm.sgu.ru: 226.
 - * Średnie: acm.uva.es: 321, 10067. spoj.sphere.pl: 135. acm.sgu.ru: 213.
 - * Trudne: acm.uva.es: 589, 10097. spoj.sphere.pl: 212.
 - DFS
 - * Łatwe: acm.uva.es: 260, 280, 459, 776.
 - * Średnie: acm.uva.es: 871, 10802. spoj.sphere.pl: 38, 372.
 - * Trudne: acm.uva.es: 10410. spoj.sphere.pl: 287.
 - Silnie spójne składowe
 - * Łatwe: acm.uva.es: 10731. spoj.sphere.pl: 51.
 - * Średnie: acm.uva.es: 247.
 - * Trudne: acm.uva.es: 125, 10510.
 - Sortowanie Topologiczne
 - * Łatwe: acm.uva.es: 124, 10305. spoj.sphere.pl: 70.
 - * Średnie: acm.uva.es: 200. acm.sgu.ru: 230.
 - * Trudne: acm.uva.es: 10319. spoj.sphere.pl: 44.
 - Acykliczność, Dwuspójne składowe, mosty, punkty artukulacji
 - * Łatwe: acm.uva.es: 315.
 - * Średnie: acm.uva.es: 10199.
 - * Trudne: spoj.sphere.pl: 185, 208.
 - Ścieżka i cykl Eulera
 - * Łatwe: acm.uva.es: 117, 10129. . acm.sgu.ru: 101.

- * Średnie: acm.uva.es: 10054. acm.sgu.ru: 121.
 - * Trudne: acm.uva.es: 10441. spoj.sphere.pl: 41. acm.sgu.ru: 286.
- Minimalne Drzewo Rozpinające
 - * Łatwe: acm.uva.es: 534, 10034. spoj.sphere.pl: 368.
 - * Średnie: acm.uva.es: 10462, 10600. spoj.sphere.pl: 30.
 - * Trudne: acm.uva.es: 10147. spoj.sphere.pl: 148. acm.sgu.ru: 206.
- Dijkstra
 - * Łatwe: acm.uva.es: 336, 10048, 10099. spoj.sphere.pl: 50.
 - * Średnie: acm.uva.es: 429, 10917. spoj.sphere.pl: 15, 119.
 - * Trudne: acm.uva.es: 10171. spoj.sphere.pl: 25, 145, 391.
- Bellman-Ford
 - * Łatwe: acm.uva.es: 104, 558.
 - * Średnie: acm.uva.es: 10746, 10806.
 - * Trudne: acm.uva.es: 10449.
- Maksymalny Przepływ
 - * Łatwe: acm.uva.es: 10092, 10330. acm.sgu.ru: 194.
 - * Średnie: acm.uva.es: 10480. acm.sgu.ru: 176.
 - * Trudne: acm.uva.es: 10546. acm.sgu.ru: 212.
- Maksymalne skojarzenie w grafie dwudzielnym
 - * Łatwe: acm.uva.es: 259, 10080, 10349, 10888. spoj.sphere.pl: 286, 373. acm.sgu.ru: 190.
 - * Średnie: acm.uva.es: 10746, 10804. spoj.sphere.pl: 203, 660. acm.sgu.ru: 210, 242.
 - * Trudne: acm.uva.es: 10615. spoj.sphere.pl: 377, 412. acm.sgu.ru: 218, 234.
- Geometria obliczeniowa:
 - Odległość punktu od prostej
 - * Łatwe: acm.uva.es: 10263, 10310.
 - * Średnie: acm.uva.es: 10709, 10011.
 - * Trudne: acm.uva.es: 10762.
 - Powierzchnia wielokąta
 - * Łatwe: acm.uva.es: 10088. spoj.sphere.pl: 55.
 - * Średnie: acm.uva.es: 10002. acm.sgu.ru: 209.
 - * Trudne: acm.uva.es: 10456, 10907.
 - Przynależność punktu, Punkty przecięcia
 - * Łatwe: acm.uva.es: 191, 10084. acm.sgu.ru: 124, 129.
 - * Średnie: spoj.sphere.pl: 102. acm.sgu.ru: 253, 198, 267.
 - * Trudne: spoj.sphere.pl: 182, 332, 272.
 - Reprezentacja okręgu w postaci trzech punktów, Wypukła otoczka
 - * Łatwe: acm.uva.es: 10078, 109, 10065.

- * Średnie: acm.uva.es: 218. spoj.sphere.pl: 26. acm.sgu.ru: 227.
 - * Trudne: acm.uva.es: 10135. spoj.sphere.pl: 228. acm.sgu.ru: 277, 290.
- Sortowanie kątowne
 - * Łatwe: acm.uva.es: 10002.
 - * Średnie: acm.uva.es: 10927.
 - * Trudne: spoj.sphere.pl: 202.
- para najbliższych punktów
 - * Łatwe: acm.uva.es: 10245.
 - * Średnie: acm.uva.es: 10750. acm.sgu.ru: 120.
- Kombinatoryka:
 - generatory transpozycji
 - * Łatwe: acm.uva.es: 10911, 10063, 146, 10098. spoj.sphere.pl: 379.
 - * Średnie: acm.uva.es: 216, 10460, 10475. spoj.sphere.pl: 399. acm.sgu.ru: 222.
 - * Trudne: acm.uva.es: 10012, 10252, 195. acm.sgu.ru: 224.
 - wszystkie podzbiory zbioru
 - * Łatwe: spoj.sphere.pl: 147.
 - * Średnie: acm.sgu.ru: 249.
 - * Trudne: acm.uva.es: 811.
- Teoria Liczb:
 - Dwumian Newtona
 - * Łatwe: acm.uva.es: 369.
 - * Średnie: acm.uva.es: 10338. spoj.sphere.pl: 78.
 - * Trudne: acm.uva.es: 10219.
 - NWW
 - * Łatwe: acm.uva.es: 10104, 10179.
 - * Średnie: acm.uva.es: 10090. spoj.sphere.pl: 62.
 - * Trudne: acm.uva.es: 10294. acm.sgu.ru: 292.
 - Odwrotność modularna, Kongruencje, Potęgowanie modulo
 - * Łatwe: acm.sgu.ru: 117.
 - * Średnie: acm.uva.es: 10006.
 - * Trudne: acm.uva.es: 10710.
 - Sito i test pierwszości
 - * Łatwe: acm.uva.es: 382, 160, 406, 543, 136. acm.sgu.ru: 113.
 - * Średnie: acm.uva.es: 294, 10168, 583, 10235. acm.sgu.ru: 116, 231.
 - * Trudne: acm.uva.es: 10139. spoj.sphere.pl: 2, 134, 288. acm.sgu.ru: 200.
 - Duże liczby

- * Łatwe: acm.uva.es: 424, 10018, 324, 10035. spoj.sphere.pl: 123, 362, 94. acm.sgu.ru: 112.
 - * Średnie: acm.uva.es: 10183, 495. spoj.sphere.pl: 54, 328. acm.sgu.ru: 111, 193, 197, 299.
 - * Trudne: acm.uva.es: 623. spoj.sphere.pl: 31, 291, 279, 115. acm.sgu.ru: 247.
- Struktury Danych:
 - Find-Union
 - * Łatwe: acm.uva.es: 10583, 10608.
 - * Średnie: acm.uva.es: 793. spoj.sphere.pl: 116. acm.sgu.ru: 174.
 - * Trudne: acm.uva.es: 10158. spoj.sphere.pl: 264, 188.
 - drzewa binarne statyczne
 - * Łatwe: acm.uva.es: 105, 231, 440, 151. spoj.sphere.pl: 227.
 - * Średnie: acm.uva.es: 10635, 10200, 10015. spoj.sphere.pl: 130, 61.
 - * Trudne: acm.uva.es: 497. spoj.sphere.pl: 57. acm.sgu.ru: 254.
 - Binarne drzewa statyczne dynamicznie alokowane
 - * Średnie: acm.uva.es: 688.
 - Wzbogacane drzewa binarne
 - * Łatwe: acm.uva.es: 10869. acm.sgu.ru: 193.
 - * Średnie: acm.uva.es: 221. spoj.sphere.pl: 382. acm.sgu.ru: 199.
 - * Trudne: acm.uva.es: 501. spoj.sphere.pl: 205. acm.sgu.ru: 263.
- Algorytmy Tekstowe:
 - KMP, Aho-Corasic, Baker
 - * Łatwe: acm.uva.es: 10010. spoj.sphere.pl: 48.
 - * Średnie: acm.uva.es: 11019, 10298. spoj.sphere.pl: 263.
 - * Trudne: acm.uva.es: 10679. spoj.sphere.pl: 413.
 - Palindromy, Drzewa Sufiksowe
 - * Łatwe: acm.uva.es: 10234.
 - * Średnie: spoj.sphere.pl: 220.
- Algebra
 - Eliminacja Gaussa (także modulo)
 - * Łatwe: acm.uva.es: 10309. acm.sgu.ru: 260.
 - * Średnie: acm.uva.es: 10524, 10109.
 - * Trudne: acm.uva.es: 10808.
 - programowanie liniowe (sympleks)
 - * Łatwe: acm.uva.es: 10817.
 - * Średnie: acm.uva.es: 10498.
 - * Trudne: acm.sgu.ru: 248.
- Dynamiki

- łatwe: acm.uva.es - zadanie 348, acm.uva.es - zadanie 674, acm.uva.es - zadanie 10003, acm.uva.es - zadanie 10081, acm.uva.es - zadanie 10131, acm.uva.es - zadanie 10198, acm.uva.es - zadanie 10259, acm.uva.es - zadanie 10271, acm.uva.es - zadanie 10304, acm.uva.es - zadanie 10482, acm.uva.es - zadanie 10529, acm.uva.es - zadanie 10811, spoj.sphere.pl - zadanie 346, spoj.sphere.pl - zadanie 365, spoj.sphere.pl - zadanie 394, spoj.sphere.pl - zadanie 402, acm.sgu.ru - zadanie 104.
- średnie: acm.uva.es - zadanie 531, acm.uva.es - zadanie 562, acm.uva.es - zadanie 10069, acm.uva.es - zadanie 10201, acm.uva.es - zadanie 10280, acm.uva.es - zadanie 10296, acm.uva.es - zadanie 10400, acm.uva.es - zadanie 10405, acm.uva.es - zadanie 10549, acm.uva.es - zadanie 10558, acm.uva.es - zadanie 10930, spoj.sphere.pl - zadanie 292, spoj.sphere.pl - zadanie 338, spoj.sphere.pl - zadanie 345, spoj.sphere.pl - zadanie 348, spoj.sphere.pl - zadanie 364, acm.sgu.ru - zadanie 183, acm.sgu.ru - zadanie 269, acm.sgu.ru - zadanie 304.
- trudne: acm.uva.es - zadanie 116, acm.uva.es - zadanie 147, acm.uva.es - zadanie 357, acm.uva.es - zadanie 366, acm.uva.es - zadanie 711, acm.uva.es - zadanie 10032, acm.uva.es - zadanie 10154, acm.uva.es - zadanie 10157, acm.uva.es - zadanie 10261, spoj.sphere.pl - zadanie 350, spoj.sphere.pl - zadanie 366, spoj.sphere.pl - zadanie 388, acm.sgu.ru - zadanie 132, acm.sgu.ru - zadanie 278.

- Zachłanne

- łatwe: acm.uva.es - zadanie 120, acm.uva.es - zadanie 10020, acm.uva.es - zadanie 10249, acm.uva.es - zadanie 10440, acm.uva.es - zadanie 10821, acm.uva.es - zadanie 10954, acm.uva.es - zadanie 10965, spoj.sphere.pl - zadanie 661.
- średnie: acm.uva.es - zadanie 10461, acm.uva.es - zadanie 10563, acm.uva.es - zadanie 10716, acm.uva.es - zadanie 10720, acm.uva.es - zadanie 10785, acm.uva.es - zadanie 10982, spoj.sphere.pl - zadanie 247, acm.sgu.ru - zadanie 259.
- trudne: acm.uva.es - zadanie 410, acm.uva.es - zadanie 714, acm.uva.es - zadanie 10382, acm.uva.es - zadanie 10665, acm.uva.es - zadanie 11006, spoj.sphere.pl - zadanie 417, acm.sgu.ru - zadanie 257.

Prosty Kurs Programowania

... czyli co należy powiedzieć ludziom, którzy pierwszy raz w życiu widzą komputer. Zajęcia PKP nie powinny w żadnym stopniu powodować zmniejszenia intensywności czy zawieszania prowadzenia zajęć właściwych pod OI (ani planowanych od bieżącego roku szkolnego - oby! - zajęć przygotowujących pod

OM). Jeżeli jest konieczność prowadzenia PKP, czyli są ludzie chętni, a nie znający języka, to zajęcia należy prowadzić równolegle w tym samym semestrze, najlepiej tak by osoby z PKP mogły nadgonić ich kolegów z „lepszym startem”. Początki zawsze są najcięższe.

- wprowadzenie do obsługi `g++`
 - niech otworzą dwa okna terminala i poćwiczą (`cp`, `mkdir`, `rm`, historia komend)
 - `vim` - wprowadzenie, `:w`, `:q`, zaznaczanie, `:c`, `:y`, `set cindent`, `syntax on` (jeśli ktoś korzysta)
 - `g++ abc.cpp -o abc -lm -static -O2`
 - program typu „Hell on world“ (`./abc`)
 - wytłumaczyć że mają sobie zainstalować system operacyjny i `g++` , powiedzieć jak to zrobić. (jeśli koniecznie muszą na win to może jakiś Dev-C++)
- Budowa Programu:
 - nagłówki: `#include<cstdio>`
 - funkcja `main` (o tym że instrukcje są wykonywane po kolei, zwracanie wartości 0)
 - deklaracja zmiennej (`int`) globalnie i lokalnie w `main`
 - najpopularniejsze typy (`int`, `char`, `bool`, `long long`, `float`, `double` też wersje `unsigned`)
 - `char` to też liczba - można zwyczajnie używać w operacjach arytmetycznych np. (`int d`, `char c`) $d = 10 \cdot (c - 'a')$
 - `printf`, `scanf` wytłumaczyć znaczenie `&` oraz symboli typów zmiennych
 - `if()` , `&&` , `||` , `!` , - warunki logiczne i bloki instrukcji w klamrach (i oczywiście operatory arytmetyczne, tak by umieli każde działanie zapisać, nie zapomnieć o podłodze z dzielenia i dzieleniu modulo)
 - pętle `for`, `while`, `do while`, `break`, `continue`
 - przekierowanie wejścia/wyjścia (`./abc < abc.in > abc.out`)
 - jako ćwiczenie korzystania z tablic: generatory listy liczb pierwszych (od n^2 do $n\sqrt{n}$), program podający liczbę słów we wpisanym ciągu (np ” mietek zjadł stefana ” zwróci 3);
- własne deklaracje
 - własne funkcje, zwracanie wartości, `inline` , przykrywanie zmiennych
 - rekurencja - jakiś prosty przykład np. Liczby Fibonacciego
 - `struct` - własna struktura i korzystanie z niej
- wskaźniki i dynamiczna alokacja pamięci

- `*` , `&` - znaczenie operatorów, przekazywanie do funkcji zmiennych przez adres
 - `new` , `delete` - tworzenie/usuwanie obiektów i tablic (także kilkumiarowych - za pomocą odpowiedniej pętli - niech sami pogłówną chwilę)
 - prawdziwa potęga wskaźników - struktura z elementem wskazującym na strukturę tego samego typu, operator `->`
 - Stos wskaźnikowy - absolutnie ważny moment - muszą rozumieć i pisać
 - Lista dwukierunkowa na wskaźnikach
- trochę technikaliów
 - operacje na bitach
 - biblioteka matematyczna (zwrócić uwagę na powolne działanie wielu funkcji - np trygonometrycznych - zamiast nich docelowo będziemy gdzie się da używać iloczynu wektorowego)