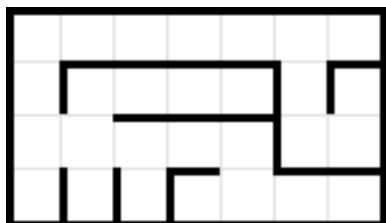


Zestaw zadań I

Witajcie, starałem się aby większość zadań uznana była za „trudne”. Jeśli zawiodłem Wasze oczekiwania to przepraszam ;) Do zestawu dołączyłem wskazówki.

1. Robot znajduje się w labiryncie złożonym z $n \times m$ jednakowych, kwadratowych pól. Pola mogą być oddzielone ścianami. Przykład (labirynt ma wymiary 7×4):



Wyobraźmy sobie, że na pozycji (x_1, y_1) stoi robot, który chce dojechać do pozycji (x_2, y_2) . Przyjmijmy, że przemieszczenie na sąsiednie pole zajmuje mu 1 sekundę. Jaki jest najkrótszy czas w jakim może dotrzeć do celu i jaką ścieżką może to zrobić?

Utrudnijmy zadanie: założmy, że robot może być skierowany w jedną z czterech stron (prawo, lewo, góra, dół), a obrót o 90 stopni zajmuje mu jedną sekundę. Założmy że w pozycji (x_1, y_1) jest skierowany do góry. Jaki ciąg operacji prowadzi go do przemieszczenia go na pozycję (x_2, y_2) .

2. Zaświecanie lampek

- Mamy rząd 8 lampek, każda może być zaświecona lub zgaszona. Jeśli zmienimy stan jakiejś lampki, to jednocześnie zmienia się stan jej istniejących bezpośrednich sąsiadów.

Przykłady:

XOOXOOXX → **O**XOXOOXX (zapalamy pierwszą lampkę)

XOO**X**OOXX → XOO**O**XXX (gasimy piątą lampkę)

XOOXOO**X**X → XOOX**O**OO (zapalamy pierwszą lampkę)

Dla każdego układu lampek określić czy istnieje ciąg gaszeń/zaświeceń który doprowadzi układ do stanu w którym wszystkie lampki są zgaszone.

Jeśli rozwiązanie korzysta z jakiegoś grafu to zastanów się jak najlepiej zaimplementować reprezentację wierzchołków tego grafu w języku C++.

- Co dzieje się gdy $n=1000000$?
3. Mamy planszę złożoną z 1001 pól. W każdym ruchu gracz może przykryć 3 sąsiednie, dotychczas nie przykryte pola, klocek długości 3. Przegrywa ten, który nie może wykonać ruchu. Czy rozpoczynający grę ma strategię wygrywającą?
 4. Mamy planszę $1 \times n$ oraz trzy rodzaje pasków: o wymiarach $1 \times n_1$, $1 \times n_2$, $1 \times n_3$ (mamy dowolną liczbę pasków danego rodzaju). Gracze na przemian kładą dokładnie jeden pasek dowolnego rodzaju na planszy (pola są zakrywane paskami w pełni – nie ma sytuacji że pasek wchodzi np. na połowę pola). Nie można kłaść paska na pole na którym już znajduje się inny pasek. Przegrywa gracz który nie może wykonać ruchu.
Podać rozwiązanie o złożoności $O(n^2)$ obliczające czy gracz rozpoczynający ma pozycję wygrywającą.
 5. Armia

- Mamy zbiór N liczb. Jak w złożoności czasowej $O(N)$ sprawdzić czy da się podzielić ten zbiór na K części tak, aby liczby w danej części liczby różniły się o co najwyżej M .
- Sytuacja podobna jak w poprzednim zadaniu. Chcemy jednak powiedzieć jakie jest najmniejsze M , że zadany zbiór N liczb da się podzielić na K części, takie że w danej części liczby różnią się o co najwyżej M .
- Zadanie z II etapu OIG - ?„Armia Napoleona”? Załóżmy że teraz mamy N przedziałów. Z każdego przedziału chcemy wybrać jedną liczbę, a następnie wybrane liczby podzielić na K części, tak, aby maksymalna różnica w obrębie danej części była jak najmniejsza.
- Jeśli dotarłeś dotąd to zrób: <http://ki.staszic.waw.pl/task.php?name=diamenty>

*) Zadanie bonusowe, dla zaznajomionych z programowaniem dynamicznym
Mając dwa słowa **wypisać** ich najdłuższy wspólny podciąg. Złożoność **pamięciowa liniowa**, czasowa kwadratowa.

Wskazówki:

1. Jakich parametrów potrzeba do aktualnego opisu stanu robota? Czy można przedstawić zadanie jako problem grafowy? Jeśli tak, to co będzie wierzchołkiem, a co krawędzią? Jaki poznany algorytm grafowy służy do wyszukiwania najkrótszych ścieżek?
2. Czy graf w którym wierzchołkami byłby stany lampek byłby jakoś pomocny?
Druga część: założmy że istnieje taki ciąg operacji: ile razy w tym ciągu będziemy bezpośrednio zmieniać stan i -tej żarówki?
3. $X \text{ xor } X = 0$. Bez względu na to jakie jest X .
4. Czy gra spełnia założenia twierdzenia Sprague-Grundy'ego?
Przyjmijmy, że gracz pierwszy położył swój pasek w środku planszy. Podzielił tym samym plansze na dwie części. Czy to co się dzieje na jednej części ma wpływ na to co dzieje się na drugiej części?
Założmy też na chwilę, że rozwiązaliśmy zadanie dla wszystkich plansz $1 \times k$ dla $k < n$.
<http://www.yucs.org/~gnivasch/cgames/spraguegrundy/index.html>
5. W pkt a należy zastanowić się nad algorytmem zachłannym. W pkt b należy się zastanowić czy istnieje sytuacja, że dla pewnego M da się podzielić zbiór, a dla jakiegoś $M' > M$ już tego się zrobić nie da. Jakim prostym algorytmem można znaleźć M , takie że dla $M-1$ się nie da, a dla M się da. Jaką to ma złożoność (jest ona inna niż w a)?
W pkt c wykorzystać technikę wypracowaną w poprzednich podpunktach. W jakiej sytuacji możemy bezkarnie przesuwać „żołnierzy” z jednej grupy do drugiej?