

Królewna Śnieżka

WWI 2024 – grupa 3
Dzień 4 – 18 sierpnia 2024

Kod zadania: **sni**
Limit pamięci: **256 MiB**



No i stało się. Królewna Śnieżka postanowiła nie czekać na pomoc od Księcia z Bajki i sama jedna zaatakowała Złą Królową. Zgodnie ze wszelkimi oczekiwaniami, nasza silna i niezależna kobieta jest już prowadzona do lochów. Pozostało już tylko liczyć na pomoc od siedmiu krasnoludków.

Dzięki swoim niecodziennym zdolnościom, udało im się dostać do środka lochów jeszcze przed wtrąceniem tam Śnieżki. Dokonali już wstępnego rozpoznania lochów – jest to spójna sieć komór połączona dwukierunkowymi korytarzami. Udało im się też podpatrzeć szyfr do lochów – jest to jedna liczba całkowita X z zakresu od 0 do $2^{60} - 1$. Muszą więc tylko dyskretnie zakodować tę liczbę w lochach, ulotnić się przed przybyciem strażników i liczyć na inteligencję białogłowy.

Ale jak tu dyskretnie zakodować tak dużą liczbę? Jedyne co mają ze sobą to ... czapeczki. Dużo czapeczek. Bardzo dużo czapeczek. Pewnie zostały im po jakichś zagadkach, albo po swoich wysokowych sesjach zdjęciowych. Czapeczki są w dwóch kolorach: czarnym i białym. Postanowili więc pozostawiać czapeczki w komorach lochów, ale – aby nie było to zbyt podejrzane – po jednej czapeczce w każdej komorze. W końcu jaki strażnik będzie wszczynał alarm tylko dlatego, że zobaczył jedną krasnoludzką czapeczkę?

Teraz kilka słów o naszej silnej i niezależnej Królewnie. Zna ona te lochy, w końcu wychowała się na tym zamku. Zna je w **dokładnie tej samej formie** co krasnoludki. Zostanie wrzucona do którejś z komnat (krasnoludki nie wiedzą do której) i zacznie eksplorację. Będzie odnotowywała kolory czapeczek w komnatach, które odwiedza. Aby jej zachowanie było jak najmniej podejrzane, chciałaby ograniczyć ilość przejść, które wykona.

Twoim zadaniem jest zaimplementowanie strategii dla krasnoludków i Królewny Śnieżki.

Biblioteka

Jest to zadanie interaktywne, to znaczy Twój program będzie porozumiewał się z biblioteką. Aby użyć biblioteki, należy załączyć nagłówek `#include "snilib.h"`.

Wymagane funkcje

Twoim zadaniem jest zaimplementowanie dwóch funkcji:

- `void Krasnoludki(int n, int m, int A[], int B[], long long X, int t)`
Jest to funkcja, która powinna reprezentować strategię krasnoludków. Parametry n ($60 \leq n \leq 10000$) i m ($1 \leq m \leq 20000$) oznaczają odpowiednio liczbę komnat i korytarzy. Tablice A i B są długości m i reprezentują one korytarze. Dla każdego i ($0 \leq i < m$) istnieje krawędź między komnatą $A[i]$ i $B[i]$. Korytarze nie tworzą pętli ani multikrawędzi. Komnaty są numerowane od 0 do $n - 1$. Parametr X jest szyfrem do lochów, który trzeba zakodować. Parametr t jest numerem podzadania, więcej o tym w sekcji *Ocenianie*. Funkcja ta powinna wywoływać dostarczaną przez bibliotekę funkcję `ZostawCzapeczke`, opisaną poniżej.
- `long long Krolewna(int n, int m, int A[], int B[], int p, int v, int t)`
Jest to funkcja, która powinna reprezentować strategię Królewny Śnieżki. Parametry n , m , A , B , t są **identyczne** co w funkcji `Krasnoludki`. Parametr p oznacza numer komnaty, w której zaczyna Królewna. Parametr v jest kolorem czapeczki w komnacie p . $v = 0$ odpowiada białej czapeczce, a $v = 1$ czarnej. Funkcja ta powinna wywoływać dostarczoną przez bibliotekę funkcję `Przejdz`, opisaną poniżej. Funkcja ta powinna zwrócić odkodowaną wartość X .

Możesz tworzyć dodatkowe funkcje pomocnicze, ale **nie implementuj** funkcji `main`. Dodatkowo funkcje `Krolewna` i `Krasnoludki` nie powinny się komunikować.

Dostarczane funkcje

Biblioteka dostarcza dwie funkcje, z których możesz korzystać:

- `void ZostawCzapeczke(int a, int b)`
Parametr a oznacza numer komnaty, a b kolor czapeczki zostawionej w tej komnacie. $b = 0$ odpowiada białej czapeczce, a $b = 1$ czarnej. Funkcja ta powinna zostać wywołana dokładnie raz dla każdej komnaty. Funkcja ta może być wykonywana wyłącznie przez funkcję `Krasnoludki`.
- `int Przejdz(int a)`
Parametr a oznacza numer komnaty, do której przejdzie Królowna i zwraca kolor czapeczki w tej komnacie (0 to biała czapeczka, 1 to czarna czapeczka). Komnata, do której przechodzi Królowna musi sąsiadować z komnatą, gdzie się obecnie znajduje. Funkcja ta może być wykonywana wyłącznie przez funkcję `Królowna`.

Błędy

Zarówno biblioteka, z której korzysta program sprawdzający, jak i ta udostępniona w zakładce "Pliki", zwracają określone błędy przy niepoprawnym korzystaniu z biblioteki, razem z werdyktem *Zła Odpowiedź*. Są to:

- `Wrong Answer[1]` - parametr a w funkcji `ZostawCzapeczke` nie jest z zakresu od 0 do $n - 1$.
- `Wrong Answer[2]` - funkcja `ZostawCzapeczke` została wywołana 2 razy z tym samym parametrem a .
- `Wrong Answer[3]` - parametr b w funkcji `ZostawCzapeczke` nie jest z zakresu od 0 do 1.
- `Wrong Answer[4]` - funkcja `ZostawCzapeczke` została wywołana inną liczbę razy niż n .
- `Wrong Answer[5]` - funkcja `Królowna` zwróciła inną wartość X niż powinna.
- `Wrong Answer[6]` - parametr a w funkcji `Przejdz` nie jest z zakresu od 0 do $n - 1$.
- `Wrong Answer[7]` - parametr a w funkcji `Przejdz` nie jest komnatą sąsiadującą z aktualną pozycją Królowny.
- `Wrong Answer[8]` - funkcja `Przejdz` została wykonana więcej niż 20000 razy.

Kompilacja na swoim komputerze

Bracia Grimm udostępnili biblioteczkę testową służącą do zabawy. Może ona różnić się od biblioteczki używanej do oceniania rozwiązań. Aby z niej korzystać, zalecana jest zmiana zawartości funkcji w pliku `sni_zle.cpp`, które należy zaimplementować.

Pliki z archiwum `sni_dla_zaw.zip` dostępnego w zakładce "Pliki" należy wypakować do folderu z kodem źródłowym programu.

Aby program się skompilował należy załączyć nagłówek `#include "snilib.h"`.

Program należy skompilować razem z biblioteką `snilib.cc`. Można to zrobić za pomocą polecenia:
`g++ twoj_program.cpp snilib.cc -o twoj_program`.

Wejście

Biblioteczka znajdująca się w zakładce "Pliki" oczekuje następującego wejścia w następującej kolejności: liczb n , m , X , p , t , będących argumentami przekazywanymi do funkcji `Królowna` i `Krasnoludki`; m par liczb a_i , b_i oznaczających istnienie korytarza łączącego komnaty a_i i b_i .

Wyjście

W przypadku biblioteczki testowej, nie trzeba nic wypisywać (biblioteka robi to sama), ale można. W przypadku biblioteki prawdziwej, nie należy nic wypisywać.

Przykład

Wyjaśnienie do przykładu: Wyjaśnienie testu przykładowego, można go znaleźć razem z testową biblioteką. W tym przykładzie dane jest 60 komnat, z czego każda para jest połączona. Dzięki temu Królewna Śnieżka może poruszać się dowolnie. Test ten spełnia podzadanie 1, dlatego $t = 1$.

Liczbą do przekazania jest 213769420, na szczęście Krasnoludki mają dobry system. Początek szyfru to 2137, zatem ustawiają wartości w komnatach numer 6 i 9 odpowiednio na 0 i 1. W tym celu wywołują `ZostawCzajeczke(6, 0)` oraz `ZostawCzajeczke(9, 1)`; Kolejny fragment to 69, dlatego ustawiają wartości w komnatach numer 16 i 19 odpowiednio na 1 i 0. W tym celu wywołują `ZostawCzajeczke(16, 1)` oraz `ZostawCzajeczke(19, 0)`; Kolejny fragment to 420, dlatego ustawiają wartości w komnatach numer 26 i 29 odpowiednio na 1 i 1. W tym celu wywołują `ZostawCzajeczke(26, 1)` oraz `ZostawCzajeczke(29, 1)`; Jest to koniec szyfru, co komunikują poprzez ustawienie wartości w komnatach numer 36 i 39 odpowiednio na 0 i 0. W tym celu wywołują `ZostawCzajeczke(36, 0)` oraz `ZostawCzajeczke(39, 0)`; Nie muszą już nic więcej przekazywać, dlatego ustawiają wartości w pozostałych komnatach na 0. W tym celu wywołują `ZostawCzajeczke(i, 0)` dla każdego i z zakresu od 0 do 59 nie wymienionego wcześniej. Po tym kończą swoją część programu.

Królewna zaczyna w komnacie nr 1, gdzie wartością jest 0. Następnie idzie odpowiednio do komnat: 6, 9, 16, 19, 26, 29, 36, 39 i odczytuje odpowiednio wartości: 0, 1, 1, 0, 1, 1, 0, 0. Aby to uzyskać wywołuje: `Przejdz(6)`, `Przejdz(9)`, `Przejdz(16)`, `Przejdz(19)`, `Przejdz(26)`, `Przejdz(29)`, `Przejdz(36)` i `Przejdz(39)`. W tym momencie wszystko jest już jasne i zwraca wartość "2137" + "69" + "420" = 213769420.

Ocenianie

| Podzadanie | Ograniczenia | Limit czasu | Liczba punktów |
|------------|---|-------------|----------------|
| 1 | $t = 1, n \leq 300$ | 6 s | 8 |
| 2 | $t = 2$ | 6 s | 10 |
| 3 | $t = 3, m = n - 1, A[i] = i, B[i] = i + 1$, funkcja <code>Przejdz</code> może zostać wywołana co najwyżej 250 razy | 6 s | 10 |
| 4 | $t = 4, 240 \leq n$ | 6 s | 55 |
| 5 | $t = 5$, funkcja <code>Przejdz</code> może zostać wywołana co najwyżej 120 razy | 6 s | 17 |

Wynik za podzadanie 4 jest obliczany następująco:

Niech C oznacza największą liczbę razy, ile Twój program wywołał funkcję `Przejdz` na tym podzadaniu.

- 0 punktów, jeśli $960 < C$.
- $\lfloor 55 - 13 \log_2(\frac{C}{120}) \rfloor$ punktów, jeśli $120 < C \leq 960$.
- 55 punktów, jeśli $C \leq 120$.