

Przygoda

Krzysztof Leszek

Grupa odważnych bohaterów postanowiła podjąć się zadania zgłędzenia złego smoka. Zanim zmierzyli się z legendarną bestią musieli przejść labirynt prowadzący do jej leża. Jako odpowiedzialni (i chciwi) poszukiwacze przygód przeszukali każdy pokój labiryntu zapisując jego położenie i zawartość na mapie. Stwierdzili że nie muszą zbierać już teraz skarbów które w nich znaleźli, ponieważ mogą się tym zająć po pokonaniu smoka, a teraz by ich jedynie spowalniały. Pewni siebie i żądni bogactw stanęli więc do walki ze smokiem. Legendarny jaszczur okazał się jednak zdecydowanie zbyt silny dla naszych bohaterów i musieli oni wycofać się na z góry upatrzone pozycje (uciekać ile sił w nogach). W tym celu muszą użyć wcześniej przygotowanej mapy labiryntu, jednakże ich wcześniejsze zwiedzanie zostawiło ich z resztką zapasu pochodni, tak więc żeby nie zgubić się w ciemnościach będą musieli zebrać pochodnie które wcześniej oznaczyli na mapie. Nasi bohaterowie przypomnieli sobie również, że aby zakupić sprzęt potrzebny na tą wyprawę zaciągnęli poważne pożyczki w Gildii Bohaterów i żeby je spłacić (i nie zginąć z rąk bankierów) muszą zebrać w labiryncie wystarczającą ilość złota. Gdy rozważali swoją sytuację, bohaterowie usłyszeli ryk smoka i dźwięk walących się tuneli, przez co wiedzieli że nie ważne jaką ścieżkę wybiorą, bestia będzie im deptała po piętach.

Aby uratować pechowych bohaterów napisz wykorzystując programowanie z nawrotami program, który znajdzie jakąkolwiek ścieżkę przez labirynt (jeżeli taka istnieje) którą bohaterowie są w stanie przejść cały czas oświetloną i zebrać wystarczająco dużo złota aby spłacić swoje długi, jednocześnie unikając smoka. Przyjmuje się że wierzchołek 0 to pierwszy pokój od leża smoka, a wierzchołek $n - 1$, gdzie n to liczba wszystkich pokoi, to wyjście. Jeżeli bohaterom skończą się pochodnie ale są w pokoju wyjścia to ścieżka liczona jest jako poprawna.

Smok idzie po ścieżce bohaterów i niszczy wszystkie pokoje po których przejdzie. Porusza się on w następujący sposób:

Etap 1 (1 pkt) Smok idzie tuż za bohaterami, przez co bohaterzy nigdy nie będą mogli wejść ponownie do wcześniej odwiedzonego pokoju.

Etap 2 (1,5 pkt) Smok potrzebuje dodatkowe czasu żeby wygrzebać się ze swojego leża, przez co przez pewną ilość ruchów bohaterów nie będzie nic robił. Ilość tych ruchów opisana jest parametrem *dragonDelay*. Po odczekaniu *dragonDelay* ruchów, smok zużywa dodatkowy ruch żeby wejść do pokoju 0 po czym zaczyna podążać za bohaterami. Smok wykonuje ruchy w tym samym momencie co bohaterowie i zawsze podąża najświeższym ich tropem, to znaczy że może iść tylko po ich śladach, a jeżeli bohaterowie odwiedzą ten sam pokój wiele razy to ruszy on w stronę w którą ostatnio wyszli z tego pokoju.

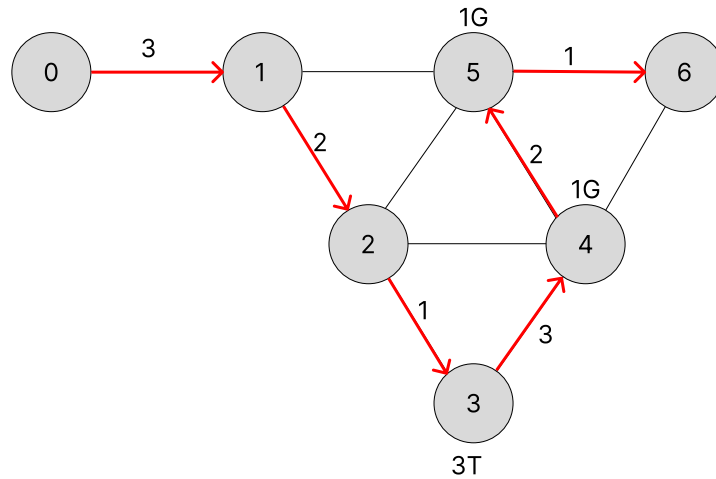
Dane

- **Graph labirynth** — graf opisujący labirynt, gdzie wierzchołki to pokoje a krawędzie między nimi oznaczają korytarze łączące te pokoje.
- **int torches** — ilość pochodni które na start posiadają bohaterowie, jedna pochodnia starczy im dokładnie na przejście dokładnie jednego korytarza i odczytanie mapy w nowym pokoju.
- **int[] roomTorches** — tablica o długości równej ilości pokoi z informacją jaka ilość pochodni jest w danym pokoju.
- **int debt** — ilość złota którą bohaterowie muszą zebrać
- **int[] roomGold** — tablica o długości równej ilości pokoi z informacją jaka ilość złota jest w danym pokoju.
- (Etap 2) **int dragonDelay** — opóźnienie z jakim wystartuje smok i zacznie burzyć pomieszczenia w labiryncie. Smok zawsze podąża najświeższym tropem bohaterów i niszczy pokoje po których przejdzie. Oznacza to że bohaterowie mogą wielokrotnie wchodzić do tego samego pokoju, póki mają przewagę nad smokiem. Smok porusza się tak samo szybko jak bohaterowie i opóźnienie jest wyrażone w ilości ruchów jakie mogą wykonać w tym czasie bohaterowie

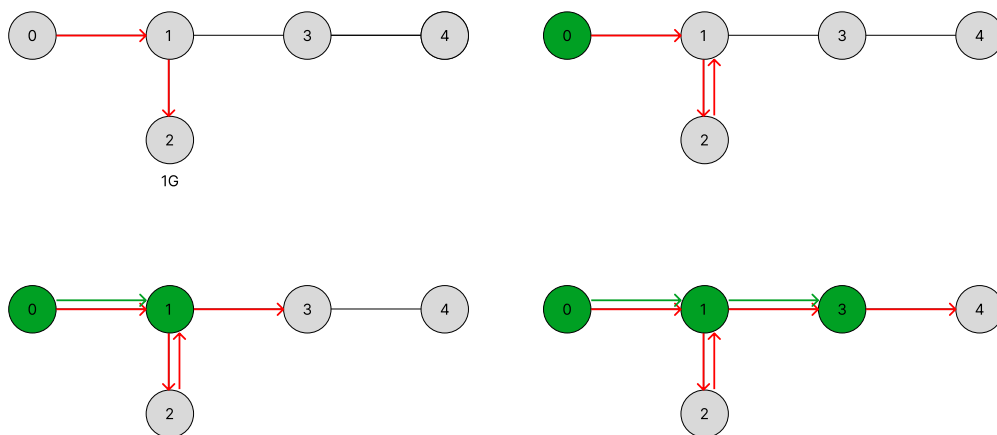
Uwagi i wskazówki

- Maksymalna ocena będzie przyznawana za rozwiązania które będą się wykonywały w parę sekund i nie będą nigdy przeglądały dwa razy tej samej trasy.
- W części domowej wymagana będzie lepsza złożoność czasowa niż w laboratoryjnej. Aby to osiągnąć należy znaleźć sposób na odrzucenie pewnych ścieżek zanim się na nie wejdzie. W zbiorze testów laboratoryjnych znajduje się jeden zakomentowany test w stylu domowych, dla ewentualnego sprawdzenia.

Przykłady



Rysunek 1: Przykład zasad zadania dla Etapu 1. Startuje się z wierzchołka 0, początkowa liczba pochodni to 3, a ilość złota do zebrania to 2. Przy wierzchołkach znajdują się informacje o złocie (G) i pochodniach (T) w tym wierzchołku. Na czerwono zaznaczono poprawną trasę, przy jej krawędziach pokazane są zapasy pochodni w momencie ruszania przez daną krawędź. Można zauważyć że limit na pochodnie nie pozwala od razu pójść do wierzchołków 4 i 5 żeby zebrać złoto, tylko trzeba wpierw odwiedzić 3 aby zebrać zapasy które wystarczą do końca.



Rysunek 2: Przykład dla Etapu 2. Początkowa liczba pochodni to 4, ilość złota do zebrania to 1, opóźnienie smoka to 2. Aby zebrać wystarczająco złota należy wejść do wierzchołka 2, ale nie ma z niego drogi dalej, tak więc przykład ten byłby niemożliwy dla Etapu 1. W Etapie 2 możemy się cofnąć z powrotem do wierzchołka 1. Jednakże, jako że samo dotarcie do wierzchołka 2 zajęło nam dwa ruchy, to w momencie gdy wchodzimy z powrotem do wierzchołka 1, to smok wkracza na wierzchołek 0 i od tego momentu będzie nas gonił. Co więcej, w momencie gdy wejdzie on do wierzchołka 1, to dalej będzie szedł prosto do wierzchołka 3, ponieważ zawsze podąża on najświeższym tropem.