

Zadanie: MIN

Miny



Warsztaty ILO 2017-2018, grupa olimpijska, dzień 15.

Pomyślmy o naszych minach jako o wierzchołkach, a o zasięgu min jako o krawędziach do min które one wysadzają po wybuchu.

Zauważmy, że jeśli na takich grafie policzymy silnie spójne składowe, to wybuch dowolnej miny z SSS spowoduje wybuch wszystkich innych min z tej SSS. Scalmy więc SSS w jeden wierzchołek i przypiszmy jej koszt równy minimalnemu kosztowi wysadzenia pewnej min w tej SSS.

Teraz mamy pewien DAG. Aby wysadzić wszystkie miny, musimy wysadzić te wierzchołki do których nic nie wchodzi. Zauważmy też, że nie musimy wysadzać nic więcej.

Wysadzamy więc po jednej minie w każdej silnie spójnej do której nic nie wchodzi.

Zauważmy, że możemy dla każdej SSS trzymać seta z aktualnymi kosztami wysadzenia wierzchołków i aktualizować go przy zapytaniu zmieniając koszt wysadzenia wszystkich min, więc obsługa zapytań jest prosta.

Rozwiązanie brutalne $O(n^2 + q \cdot \log(n))$

Najprostszym sposobem jest jawne skonstruowanie grafu i policzenie silnie spójnych za pomocą standardowych algorytmów.

Rozwiązanie wzorcowe $O(n \cdot \log(n) + q \cdot \log(n))$

Najpierw zauważmy, że krawędzie wychodzące z każdego z wierzchołków tworzą spójny przedział.

Będziemy więc chcieli policzyć SSS na grafie "przedziałowym".

Możemy nad ciągiem min zbudować drzewo przedziałowe i skierować jego krawędzie w dół. Jeśli z jakiegoś wierzchołka jest krawędź do przedziału, możemy zamiast do każdego wierzchołka z przedziału dać z niego krawędzie jedynie do przedziałów bazowych. W ten sposób da się dojść do każdego wierzchołka z przedziału.

Na takim grafie możemy obliczyć SSS standardowym sposobem, bo ma mały rozmiar. Można też łatwo stwierdzić, do których SSS nic nie wchodzi, trzeba tylko "wyłączyć", drzewowe wierzchołki do których nie da się dojść z żadnej miny.

Rozwiązanie alternatywne $O(n \cdot \log(n) + q \cdot \log(n))$

Możemy tutaj zastosować algorytm obliczania SSS za pomocą funkcji low. Zauważmy, że da się go przyspieszyć. Co robimy z krawędziami, które przeglądamy obliczając funkcję low?

- jeśli wierzchołek jest nie odwiedzony wywołujemy na nim dfsa i minujemy jego z naszym wynikiem – to można rozwiązać łatwo trzymając wierzchołki nieodwiedzone na secie
- jeśli wierzchołek jest już przetworzony, ale nie ma przypisanej silnie spójnej minujemy wynik z jego czasem wejścia – można to zrobić na drzewie przedziałowym
- jeśli wierzchołek ma już przypisaną SSS, nie robimy z nim nic – można po przypisaniu mu numeru SSS dodać na drzewo nieskończoność

Przypomnę, że wierzchołkom przypisujemy numer SSS wtedy gdy w wierzchołku x $low[x] == in[x]$. Zdejmujemy wtedy wszystkie wierzchołki ze stosu aż nie zdejmujemy wierzchołka x (wrzucamy na stos przy wchodzeniu do wierzchołka)

Mając już podział na SSS trzeba jeszcze policzyć do jakich SSS nic nie wchodzi. Warto tutaj zauważyć, że SSS nie muszą być przedziałami. Tą część też można zrobić np. drzewiem przedziałowym, choć za pewno da się prościej.

