

Park wodny

Park wodny Aquabajt bierze udział w konkursie na największy basen. Teren parku, na którym zlokalizowane są baseny, ma kształt kwadratu o boku długości n i jest podzielony na n^2 segmentów, z których każdy jest kwadratem o boku długości 1. Każdy z segmentów może być basenikiem albo alejką między basenikami. Baseniki połączone bezpośrednio ze sobą (czyli będące segmentami stykającymi się bokami) tworzą większe baseny. Obecnie w parku wodnym każdy basen ma kształt prostokąta.

Dyrekcja Aquabajtu postanowiła zwiększyć swoje szanse na wygraną w konkursie, przebudowując park. Ze względu na ograniczony czas i fundusze zdecydowano o przekształceniu co najwyżej dwóch segmentów z alejek na baseniki. Pomóż władzom parku uzyskać basen złożony z maksymalnej liczby baseników. Zakładamy, że po przebudowie największy basen nie musi być już prostokątem.

Wejście

Pierwszy wiersz standardowego wejścia zawiera jedną dodatnią liczbę całkowitą n oznaczającą wielkość parku wodnego.

W następnych n wierszach znajduje się dwuwymiarowa mapa parku: każdy z tych wierszy zawiera słowo złożone z n liter. Litera A oznacza segment z alejką, natomiast litera B oznacza basenik. Możesz założyć, że w opisie znajduje się co najmniej jedna litera B.

Wyjście

Pierwszy i jedyny wiersz standardowego wyjścia powinien zawierać jedną liczbę całkowitą oznaczającą wielkość największego basenu, jaki można uzyskać.

Przykład

Dla danych wejściowych:

5

BBBAB

BBBAB

AAAAA

BBABA

BBAAB

poprawnym wynikiem jest:

14

Testy „ocen”:

1ocen: $n = 10$, tylko jeden basen na planszy,

2ocen: $n = 10$, cała plansza pokryta basenem,

3ocen: $n = 1000$, szachownica.

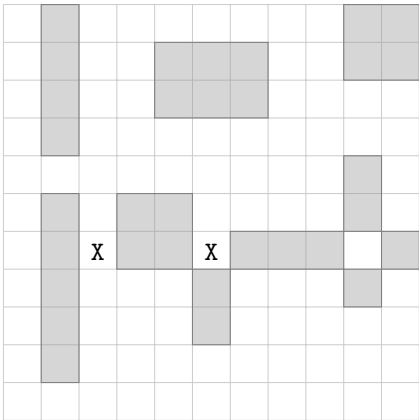
Ocenianie

Zestaw testów dzieli się na następujące podzadania. Testy do każdego podzadania składają się z jednej lub większej liczby osobnych grup testów.

Podzadanie	Warunki	Liczba punktów
1	$n \leq 10$	11
2	$n \leq 50$, liczba basenów na początku nie przekroczy 80	11
3	$n \leq 60$	22
4	$n \leq 1000$, na początku każdy basen jest prostokątem 1×1	22
5	$n \leq 1000$	34

Rozwiązanie

W zadaniu mamy daną mapę podzieloną na n^2 segmentów. Każdy z segmentów może być basenikiem albo alejką między basenikami. Baseniki połączone bezpośrednio ze sobą tworzą większe baseny, które są początkowo prostokątami (zaznaczone na rysunku kolorem szarym). Chcielibyśmy zamienić dwie alejki na baseniki, tak aby utworzyć jak największy basen (który nie musi być już prostokątem).



W optymalnym rozwiązaniu dla powyższego przykładu zbudujemy dwa baseniki oznaczone znakiem X, otrzymując jeden basen złożony z $5 + 1 + 4 + 1 + 2 + 3 = 16$ baseników.

Rozwiązanie brutalne $O(n^6)$

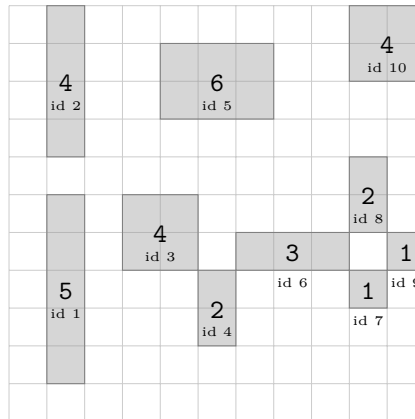
Sprawdzamy każdą możliwą parę segmentów i wybieramy tę, której zamiana na baseniki utworzy największy basen. Aby znaleźć największy basen po takiej zamianie, możemy wyobrazić sobie graf, w którym wierzchołkami są segmenty zawierające baseniki, a krawędzie łączą wierzchołki odpowiadające sąsiednim segmentom, a następnie

użyć algorytmu przeszukiwania grafu w głąb (DFS). Jako że wszystkich par segmentów jest $O(n^4)$, natomiast znalezienie największego basenu zajmuje czas liniowy względem liczby segmentów, złożonością czasową tego rozwiązania jest $O(n^6)$. Warto zwrócić uwagę na przypadek szczególny (patrz test `2ocen`), w którym cały park jest od początku pokryty basenem, więc nie ma żadnego segmentu niebędącego basenikiem (natomiast nie jest możliwe, żeby początkowo tylko jeden segment nie był basenikiem).

Rozwiązanie to zaimplementowane jest w pliku `pars1.cpp`. Za poprawne zaprogramowanie takiego rozwiązania na zawodach można było uzyskać około 10% punktów.

Rozwiązanie wolne $O(n^4)$

Na początku obliczamy wielkość wszystkich basenów w czasie $O(n^2)$, zapisując dla każdego basenu jego identyfikator i rozmiar. Można to zrobić na kilka sposobów, np. używając opisanego wcześniej grafu.



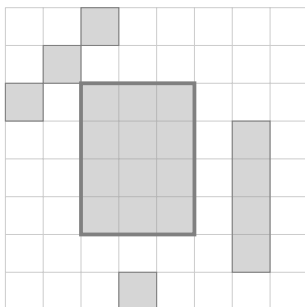
W praktyce informacje te najwygodniej zapisać w każdym segmencie basenu. Następnie rozpatrujemy każdą możliwą parę segmentów i sprawdzamy, które baseny zostaną połączone, gdy te segmenty zmienimy w baseniki. Zauważmy, że każdy segment sąsiaduje z co najwyżej czterema basenami, więc dodanie dwóch segmentów powoduje połączenie tylko stałej liczby basenów. Jako że znamy identyfikatory basenów i ich rozmiary, to w czasie stałym obliczamy całkowity rozmiar powstałego basenu.

Implementacja takiego rozwiązania znajduje się w plikach `pars3.cpp` i `pars9.cpp`. Rozwiązanie tego typu otrzymywało na zawodach około 40% punktów.

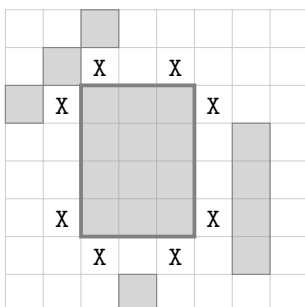
Rozwiązanie wzorcowe $O(n^2)$

Załóżmy na początek, że zmieniana para segmentów sąsiaduje ze sobą. Możemy rozważyć wszystkie takie pary, ponieważ będzie ich tylko $O(n^2)$. Dla każdej pary w czasie stałym sprawdzamy jak wyżej, które baseny zostaną połączone, i zapamiętujemy najlepszy wynik.

Dalej zakładamy, że zmieniane segmenty nie będą ze sobą sąsiadować. Zauważmy, że w tym przypadku w optymalnym rozwiązaniu segmenty te będą sąsiadować z jednym, wybranym basenem. Każdy taki basen będziemy rozpatrywali oddzielnie. Przykładowo, na rysunku poniżej rozważmy basen zaznaczony pogrubioną linią.



Znajdźmy wszystkie segmenty, za pomocą których możemy rozbudować wybrany basen. Liczba takich segmentów będzie liniowa względem obwodu rozpatrywanego basenu. Prawie wszystkie segmenty łączą się z co najwyżej jednym innym basenem. Jedyne segmenty, które mogą się łączyć z dwoma (lub nawet trzema) basenami, znajdują się na rogach. Dokładniej, są po dwa takie segmenty na każdym z rogów, co daje łącznie maksymalnie osiem różnych segmentów (oznaczonych na rysunku znakiem X).



Niech k oznacza obwód rozważanego basenu. Wszystkie narożne segmenty rozpatrujemy ze wszystkimi innymi segmentami (również tymi, które nie są na rogach), co daje maksymalnie $8k$ par segmentów. Natomiast z pozostałych segmentów (które nie są na rogach) wybieramy te dwa, które łączą się z największymi różnymi basenami – można je znaleźć w czasie $O(k)$.

Jako że suma obwodów wszystkich basenów nie przekroczy $O(n^2)$, otrzymujemy rozwiązanie wzorcowe, działające w czasie kwadratowym. Przykładową implementację można znaleźć w plikach `par.cpp` i `par2.cpp`.

Zawody II stopnia

opracowania zadań

