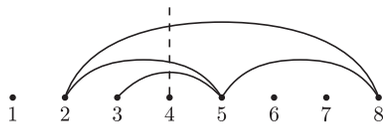
Informatyczny kącik olimpijski (10) – słownik przekątnych

W tym odcinku zajmiemy się zadaniem, które zamiast na zawodach, pojawiło się na... kolokwium (z Algorytmów i Struktur Danych). Stanowiłoby ono bardzo ładne zadanie na zawodach.



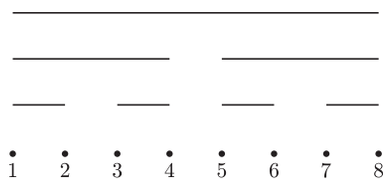
Rys. 1. Kilka przekątnych ośmiokąta reprezentowanych jako łuki. Wartością $g(4)$ jest łuk $(3, 5)$.

Weźmy wypukły n-kąt. Na początku żadna przekątna nie jest narysowana. Zadaniem jest opracowanie struktury danych, która umożliwi optymalne obsłużenie zapytań:

• **Sprawdź**(x, y) — sprawdza, czy między punktami x i y jest narysowana przekątna,

• **Dodaj**(x, y) — rysuje przekątną pomiędzy punktami x i y — o ile *nie przecina ona żadnej innej narysowanej przekątnej*,

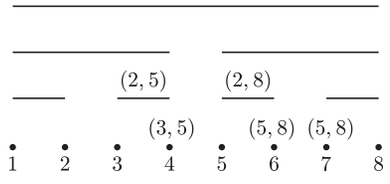
• **Usuń**(x, y) — usuwa przekątną między punktami x i y.



Rys. 2. Tak wyobrażamy sobie drzewo przedziałowe.

Pochyłą czcionką zaznaczona jest, oczywiście, główna trudność zadania (gdyby nie ten warunek, wystarczyłaby dowolna implementacja słownika).

Od czego zacząć? Na początek może ułatwimy sobie zadanie — zauważmy, że wielokąt z zadania można „wygiąć" na płaszczyźnie, aż wierzchołki utworzą ciąg punktów na prostej, ponumerowanych od 1 do n, a przekątne zamienią się na łuki (jak na rysunku 1). Resztę zadania łatwo przetłumaczyć do tej nowej wersji. W tym momencie już pojawia się pomysł — może uda się użyć drzewa przedziałowego? Ale zanim napiszemy jak, wyjaśnijmy sobie, co będziemy rozumieć przez drzewo przedziałowe.



Rys. 3. Zawartość drzewa przedziałowego po wstawieniu przekątnych z rysunku 1.

Drzewo przedziałowe (rys. 2) służy do przechowywania informacji związanych z podprzedziałami zadanego przedziału $[a, b]$. W korzeniu drzewa jest zapisana informacja związana z całym przedziałem, w jego dwóch synach informacje związane z lewą i prawą połową tego przedziału i tak dalej, aż wreszcie w liściach zapamiętywane są wartości odpowiadające pojedynczym punktom. Wartość przechowywaną w wierzchołku v będziemy oznaczać $f(v)$.

Jako ćwiczenie Czytelnik może w oparciu o drzewo przedziałowe skonstruować strukturę danych do obsługi dynamicznie zmieniającego się ciągu $a_1, \ldots, a_n$, dopuszczającą operacje (1) inicjuj wszystkie wyrazy na 0, (2) ustaw $a_i := x$ oraz (3) podaj sumę podciągu $a_i + a_{i+1} + \ldots + a_j$. Operacje (2) i (3) powinny działać w czasie $O(\log n)$.

No dobrze, ale jak to zastosować do naszego problemu?

Zapytajmy, kiedy w nowym modelu (tym z prostą), można połączyć dwa punkty x i y przekątną? Spójrzmy na rysunek 1. Dla danego t niech $g(t)$ oznacza *najniższy* łuk, który przecina pionowa półprosta wychodząca z punktu t (dotknięcie nie liczy się jako przecięcie; wartością $g(t)$ może być „pusty" łuk). Łuk od x do y można dodać dokładnie wtedy, gdy $g(x) = g(y)$. Musimy więc szybko obliczać wartości funkcji g. Do tego przyda się drzewo przedziałowe.

Przypuśćmy, że wiemy już, iż możemy dodać przekątną (x, y). Jak to zrobić? Otóż przekątna „pokryje" podciąg wierzchołków $[x + 1, y - 1]$. Słowo *pokryje* oznacza, że półprosta wychodząca z każdego z wierzchołków z tego przedziału przetnie naszą przekątną, choć niekoniecznie jako pierwszą.

Funkcja dodająca przekątną (x, y) będzie działać rekurencyjnie. Najpierw wywołujemy ją dla korzenia. Dodając przekątną (x, y) w wierzchołku v drzewa przedziałowego, wykonujemy operacje:

• jeśli podprzedział z wierzchołka v zawiera się w $[x + 1, y - 1]$, to ustaw $f(v)$ na niższy z łuków $f(v)$ i (x, y) (wysokość łuku można szybko obliczyć, jest to $y - x$),

• jeśli podprzedział z wierzchołka v jest rozłączny z $[x + 1, y - 1]$, to nic nie rób,

• w przeciwnym razie wywołaj funkcję *dodaj* dla dzieci wierzchołka v.

Łatwo zauważyć, że $g(t)$ to najniższy spośród łuków zapamiętanych w wierzchołkach drzewa odpowiadających przedziałom pokrywającym punkt t, czyli na ścieżce prowadzącej z korzenia do wierzchołka t. Zatem $g(t)$ można znaleźć w czasie $O(\log n)$.

Funkcja *Sprawdź* może działać na podobnej zasadzie, sprawdzając, czy w odwiedzanych po drodze wierzchołkach jest zapamiętana poszukiwana przekątna. (Każda wstawiona przekątna zostaje gdzieś zapamiętana, dlaczego?)

Aktualizację stanu struktury podczas usuwania przekątnej pozostawiamy Czytelnikowi.

Filip WOLSKI