

Zadania nr 2

1. Mamy ciąg n liczb całkowitych. Znaleźć podciąg spójny o największej sumie.
2. Znaleźć najdłuższą (w sensie sumy wag krawędzi) ścieżkę w grafie acyklicznym.
Rozwiązać zadanie dla grafu skierowanego i nieskierowanego. Wagi grafu są dowolnymi liczbami. Znaleźć drugą co do długości ścieżkę w takim grafie.
3. Zapoznać się z problemem odległości edycyjnej (jest na Wikipedii)
4. Zrobić zadanie: <http://ki.staszic.waw.pl/task.php?name=diamenty>
5. Mamy tort z wisienkami podzielony na N kawałków. Każdy kawałek tortu zawiera pewną liczbę wisienek. Bolek i Lolek jedzą na przemian tort po jednym kawałku i każdy z nich stara się zmaksymalizować liczbę zjedzonych przez siebie wisienek. Zaczynający grę Bolek może w swoim pierwszym ruchu zjeść dowolny kawałek z tortu, ale w następnych ruchach obaj chłopcy muszą zjeść jeden z dwóch kawałków sąsiadujących z dziurą w torcie. Jaka największą liczbę wisienek jest w stanie zjeść Bolek, przy założeniu, że obaj jedzą tort optymalnie? :)

Wskazówki:

1. Złożoność rozwiązania to $O(n)$. Rozważajmy po kolei sumy wszystkich elementów ciągu od pierwszego do i -tego. Co się dzieje w dla pierwszej takiej pozycji i , gdy ta suma jest ujemna?
2. **Przypadek skierowany:** Co może nam dać posortowanie topologiczne grafu? W jaki sposób możemy określić najdłuższą ścieżkę wierzchołka v , jeśli znamy najdłuższe ścieżki jego poprzedników?
Przypadek nieskierowany: czy mamy tu do czynienia z drzewem? Jak przerobić algorytm z przypadku skierowanego?
3. Zadanie nie wymaga wskazówki ;)
4. Rozwiązanie polegające na posortowaniu diamentów względem malejącego stosunku (v/w) i wyboru k pierwszych diamentów jest niepoprawne. Jak można zaimplementować takie błędne rozwiązanie w oczekiwanym czasie liniowym? Podaj kontrprzykład który pokazuje że to rozwiązanie jest błędne. Jeśli komuś się nudzi to polecam zapoznać się z algorytmem „mediana median”/”magicznych piątek”. Ale do rzeczy :) Na ostatnim kółku omówiliśmy technikę „wyszukiwanie binarne po wyniku”. Jak możemy – wykorzystując algorytm sortowania – odpowiedzieć na pytanie czy da się wybrać k kamieni których wartość jest większa od zadanej wartości w ?
5. Podstawową ideą programowania dynamicznego jest podział na odpowiednie podproblemy. Załóżmy że mamy obliczone kto ile zjada wisienek dla każdego k kolejnych kawałków. Jak nam to pomoże obliczyć ile kto zjada wisienek dla każdego $(k+1)$ sąsiednich kawałków?