Zliczanie, sumy prefiksowe, gąsienica, bisekcja.

1. Napisz funkcję `int* zsumuj(int n, int a[])`, która dla danej niemalejącej tablicy dodatnich liczb całkowitych $a = \{a_1, \ldots, a_n\}$ stworzy tablicę $\{b_1, \ldots, b_n\}$, gdzie $b_i = \sum_{k=a_i}^{n} a_k$. (Pamiętaj o tym, że jeśli $m > n$, to $\sum_{k=m}^{n} a_k = 0$.)
2. Dany jest niepusty wektor liczb całkowitych `v = {x_1, \dots, x_n}` zawierająca ciąg ściśle monotoniczny, to jest rosnący lub malejący. Napisz procedurę `int blisko_zera(std::vector<int> v)`, która zwraca $\min(|x_1|, \ldots, |x_n|)$.
3. (*) Dana jest tablica n liczb całkowitych $\{t_0, t_1, \ldots, t_{n-1}\}$ uporządkowana niemalejąco. Napisz procedurę: `int malo(int n, int t[])`, która dla danej tablicy t obliczy:

   $$\min\{|t_i + t_j| : 0 \leq i < j \leq n-1\}$$

   Na przykład, `malo(8, {-42, -12, -8, -1, -1, 5, 15, 60}) = 2`.
   Możesz założyć, że dana tablica t zawiera przynajmniej dwa elementy.
4. Dany jest niepusty, rosnący wektor liczb całkowitych $v = \{x_1, \ldots, x_n\}$. Napisz funkcję: `int fit(int c, const std::vector<int> v`, która dla danej liczby c i wektora v obliczy:

   $$\min\{|c + x_i + x_j| : 1 \leq i, j \leq n\}$$

   Na przykład, `fit(42, {-28, -25, -15, -1, 4, 8, 15, 60}) == 1`, ponieważ $|42 - 28 - 15| = 1$.
5. Dana jest tablica liczb całkowitych $a = \{x_1, x_2, \ldots, x_n\}$. Tablica ta ma taką własność, że jej ciąg różnicowy $((x_{i+1} - x_i)_{i=1,2,\ldots,n-1})$ jest ściśle rosnący.
   1. [PCh] Napisz procedurę `bool rozne(int n, int a[])`, która dla danej tablicy sprawdza, czy jej elementy są parami różne.
   2. Napisz procedurę `int minimum(int n, int a[])`, która dla danej tablicy wyznacz minimum z jej elementów.
6. Dana jest tablica par dodatnich liczb rzeczywistych (`float`). Liczby te, to długości boków pewnych prostokątów o takich samych polach. Tablica ta jest posortowana rosnąco po pierwszych elementach par, oraz malejąco po drugich elementach par. Napisz procedurę `float diagonal(int n, float a[])`, która wyznaczy najkrótszą z przekątnych prostokątów.
7. (*) [Bentley]
   Dana jest tablica n liczb całkowitych t. Należy znaleźć takie $0 \leq a \leq b < n$, że suma $\sum_{i=a}^{b} t[i]$ jest największa. Napisz funkcję `int p(int n, int t[])`, że $\mathrm{p}(n, t) = \sum_{i=a}^{b} t[i]$ (dla a i b jak wyżej).
8. Dana jest tablica rozmiaru $n + 1$ zawierająca liczby od 1 do n. Napisz procedurę, która zwraca (dowolną) wartość powtarzającą się w tablicy.
   Twoja procedura powinna działać w stałej pamięci dodatkowej. Natomiast może modyfikować daną tablicę.
9. Dana jest tablica liczb całkowitych zawierająca permutację liczb całkowitych od 0 do n (dla $n \geq 0$). Napisz procedurę `int cykl(const std::vector<int> p)`, która wyznacza długość najdłuższego cyklu danej permutacji.
   Na przykład: `cykl({2, 1, 0, 5, 6, 4, 3, 8, 7}) == 4`
10. (*) Mamy daną prostokątną tablicę $x\_size \times y\_size$ wypełnioną liczbami całkowitymi (x_size i y_size to stałe dodatnie), której wiersze są uporządkowane ściśle rosnąco, a kolumny ściśle malejąco. Napisz procedurę `int znajdz(int tab[y\_size][x\_size], int v)`, która policzy ile razy zadana wartość v pojawia się w danej tablicy.

?

11. (*) Dana jest tablica liczb całkowitych, która była posortowana ściśle rosnąco, a potem poddano ją rotacji cyklicznej o pewną liczbę miejsc. Napisz procedurę `int find(int arr[], int size)`, która wyznaczy o ile pozycji tablica została przerotowana -- innymi słowy, znajdzie indeks najmniejszej wartości w tablicy.
12. Dana jest tablica nieujemnych liczb całkowitych. Napisz procedurę `int find(unsigned int a[], unsigned int size)`, która sprawdza, czy w danej tablicy `a` istnieje taki spójny fragment, którego suma jest równa zadanej liczbie s (inaczej mówiąc, czy istnieją takie indeksy i i j, że $a[i] + \cdots + a[j] = s$. Jeżeli tak, wynikiem powinno być i. Jeżeli nie, wynikiem powinno być -1.
13. Dana jest funkcja nierosnąca $f : \texttt{int} \to \texttt{int}$, taka, że $f(x+1) \geq f(x) - 1$. Napisz funkcję, która znajduje punkt stały f, tzn. taką liczbę x, że $f(x) = x$ (lub jeśli punkt stały nie istnieje, to taką liczbę x, że $f(x-1) \geq x$ oraz $f(x) \leq x$).
14. Dana jest tablica liczb całkowitych $a = \{x_1, x_2, \ldots, x_n\}$. Napisz procedurę `int daleko(int n, int a[])`, która wyznaczy taką parę indeksów (i, j), że:
    - $i < j$,
    - $a.(i) < a.(j)$,
    - $j - i$ jest maksymalne.

    Wynikiem powinna być różnica $j - i$.
15. (*) Dana jest dwuwymiarowa tablica a wypełniona nieujemnymi liczbami całkowitymi. Reprezentuje ona mapę prostokątnego lasu, a wartości tablicy odpowiadają liczbie drzew rosnących w różnych miejscach. Napisz procedurę `int prostokąt(const vector<int> &a, int k)`, która dla zadanej tablicy a oraz liczby k znajdzie w obrębie tablicy (lasu) taki prostokąt, który:
    - zawiera przynajmniej k drzew, tzn. suma elementów tablicy w prostokącie jest $\geq k$,
    - obwód prostokąta jest minimalny.

    Wynikiem procedury powinien być obwód tego prostokąta. Podaj złożoność czasową i pamięciową swojego rozwiązania.
    Prostokąt o dolnym lewym rogu (x_1, y_1) i górnym prawym (x_2, y_2) zawiera wszystkie elementy tablicy $a[i][j]$ dla $x_1 \leq i \leq x_2, y_1 \leq j \leq y_2$, oraz ma obwód $2 \cdot (x_2 - x_1 + 1) + 2 \cdot (y_2 - y_1 + 1)$.
16. Dana jest prostokątna mapa złóż gazu. Mapa jest dana w postaci dwuwymiarowej prostokątnej tablicy nieujemnych liczb całkowitych, oznaczających ilość znajdującego się w danym miejscu gazu. Chcemy zakupić prawa do wydobywania gazu z terenu, który ma kształt prostokąta i zawiera łącznie przynajmniej k jednostek gazu. Napisz procedurę `int gaz(const vector<vector<int>> &m, int k)`, która obliczy minimalną powierzchnię takiego terenu.
17. (*) Zosia bardzo lubi czekoladę z orzechami, a im więcej orzechów tym lepiej. Mama pozwoliła Zosi zjeść jeden prostokątny kawałek czekolady składający się z dokładnie k kosteczek. Zosia wpatruje się w tabliczkę czekolady i zastanawia się jaki kawałek wybrać, tak żeby było w nim jak najwięcej orzechów. Kawałek ten może być wybrany z dowolnego miejsca w czekoladzie, nawet ze środka tabliczki. Pomóż Zosi.
    Napisz procedurę `int orzechy(int k, const vector<vector<int>> & czek)`, która mając dane: liczbę k ($k > 0$) oraz prostokątną tablicę nieujemnych liczb całkowitych, określającą ile orzechów znajduje się w poszczególnych kosteczkach tabliczki czekolady, wyznaczy maksymalną liczbę orzechów w kawałku czekolady, który może zjeść Zosia.
    Możesz założyć, że tabliczka czekolady jest na tyle duża, że zawiera przynajmniej jeden prostokątny kawałek czekolady złożony z dokładnie k kosteczek.
    Na przykład, dla $k = 6$ oraz tablicy:

0	0	4	1	2	0
3	3	8	11	3	2
1	3	9	8	1	2
2	1	1	2	0	12

    poprawnym wynikiem jest 42.
18. Mamy most linowy długości n metrów ($n \geq 1$). Dla każdego metrowego odcinka mostu znamy jego wytrzymałość, to jest maksymalny ciężar jaki może znaleźć się

łącznie na moście nie powodując rozerwania lin na tym odcinku. Możemy rozmieścić k podpór, dzieląc efektywnie most na $k+1$ przylegających do siebie połączonych mostów. Podpory mają znikomą szerokość i rozmieszczamy je na styku metrowych odcinków mostu.

Przed wjazdem na most zostaną ustawione dwa ograniczenia: maksymalny dopuszczalny ciężar pojazdu c jaki może wjechać na most, oraz minimalna odległość d jaką należy zachować między pojazdami. Jeśli dwie kolejne podpory znajdują się w odległości x, to odcinki łączącego je mostu muszą mieć wytrzymałość przynajmniej $\lceil \frac{x}{d} \rceil \cdot c$.

Ograniczenie c będzie równe minimalnej wytrzymałości całego mostu. Twoim zadaniem jest wyznaczenie minimalnej możliwej wartości ograniczenia d, jakie można uzyskać odpowiednio rozmieszczając podpory. Napisz procedurę `int most(const vector<int> & m, int k)`, która mając dane wytrzymałości odcinków mostu oraz liczbę podpór k wyznaczy minimalną możliwą wartość ograniczenia d.

Last modified: Saturday, 21 January 2023, 12:28 PM