



# Сума на минимални елементи

locked

Problem

Submissions

Leaderboard

Discussions

Дадени са ви  $N$  цели числа  $a_0, \dots, a_{N-1}$ , и цяло число  $d$ . Намерете сумата от минималните елементи на всички подмасиви с дължина  $d$ :

$$\bullet \sum_{i=0}^{N-d} \min(a_i, a_{i+1}, \dots, a_{i+d-1}).$$

## Input Format

Първият ред на стандартния вход съдържа две цели числа  $N$  и  $d$  - броя на числата и дължината на подмасивите.

Следват  $N$  на брой цели числа - стойностите на  $a_0, \dots, a_{N-1}$ .

## Constraints

$$0 \leq N \leq 10^6$$

$$0 \leq d \leq N$$

$$-10^9 \leq a_i \leq 10^9$$

## Output Format

Изведете едно цяло число - търсената сума.

## Sample Input 0

```
10 4
1 3 9 5 2 7 4 11 5 6
```

## Sample Output 0

```
17
```

## Explanation 0

Минималните елементи във всички подмасиви с дължина 4 са съответно: 1, 2, 2, 2, 2, 4, 4. Като ги съберем получаваме отговора 17.

[f](#) [t](#) [in](#)

Submissions: 93


Max Score: 100

Difficulty: Hard

Rate This Challenge:

☆☆☆☆☆

[More](#)

Current Buffer (saved locally, editable)  

C++14



```
1 #include <cmath>
2 #include <cstdio>
3 #include <vector>
4 #include <iostream>
5 #include <algorithm>
6 using namespace std;
7
8 vector<int> days;
9
10 struct Node
11 {
12     int data;
13     Node* next = nullptr;
14     Node* prev = nullptr;
15
16     Node(int data, Node* next = nullptr, Node*prev=nullptr)
17     {
18         this->data = data;
19         this->next = next;
20         this->prev = prev;
21     }
22 };
23
24 class Queue
25 {
26 private:
27     Node* head, * tail;
28     int length;
29     int minEl;
30 public:
31     Queue()
32     {
33         head = nullptr;
34         tail = nullptr;
35         length = 0;
36         minEl = -1;
37     }
38
39     Node* getHead() { return head; }
40     Node* getTail() { return tail; }
41     int size() { return length; }
42     void setHead(Node* head) { this->head = head; }
43     void setTail(Node* tail) { this->tail = tail; }
44
45     void push_back(int data)
46     {
47         if (head == nullptr)
48         {
49             head = new Node(data);
50             tail = head;
51             length++;
52         }
53         else
54         {
55             Node* el = new Node(data);
56             tail->next = el;
57             el->prev = tail;
58             tail = tail->next;
59             length++;
60         }
61     }
62
63     void push_front(int data)
64     {
65         if (head == nullptr)
66         {
67             head = new Node(data);
68             tail = head;
69             length++;
70         }
71         else
72         {
```

```
73         Node* el = new Node(data);
74         el->next = head;
75         head->prev = el;
76         head = el;
77         length++;
78     }
79 }
80
81 void pop_front()
82 {
83     if (head == nullptr) return;
84     if (length==1)
85     {
86         delete head;
87         head = nullptr;
88         tail = nullptr;
89         length = 0;
90         return;
91     }
92     Node* temp = head;
93     head = head->next;
94     head->prev = nullptr;
95     delete temp;
96     length--;
97 }
98
99 void pop_back()
100 {
101     if (head == nullptr) return;
102     if (head == tail)
103     {
104         delete head;
105         head = nullptr;
106         tail = nullptr;
107         length = 0;
108         return;
109     }
110     Node* temp = tail;
111     tail = tail->prev;
112     tail->next = nullptr;
113     delete temp;
114     length--;
115 }
116
117 Node* at(int ind)
118 {
119     if (ind == 0) return head;
120     int count = 0;
121     Node* res = head;
122     while (count != ind)
123     {
124         res = res->next;
125         count++;
126     }
127     return res;
128 }
129
130 void print()
131 {
132     Node* i = head;
133     while (i)
134     {
135         cout << i->data << ' ';
136         i = i->next;
137     }
138 }
139
140 Node* min()
141 {
142     if (length == 0) return nullptr;
143     if (head == tail) return head;
144
145     Node* min = head;
146     Node* current = head->next;
147
```

```
148     while (current)
149     {
150         if (min->data > current->data)
151             min = current;
152         current = current->next;
153     }
154
155     return min;
156 }
157 };
158
159
160 int main() {
161     int n, d, el;
162     cin >> n >> d;
163
164     Queue q;
165     Queue dq;
166     vector<int> v;
167
168     for (int i = 0; i < n; i++)
169     {
170         cin >> el;
171         v.push_back(el);
172     }
173
174     if (d == 0)
175     {
176         cout << 0;
177         return 0;
178     }
179
180     long long int sum = 0;
181
182     int i = 0;
183     for (; i < d; i++) //fill the queue with the first d elements
184     {
185         if (q.size() == 0)
186         {
187             q.push_back(v[i]);
188             dq.push_back(v[i]);
189         }
190         else
191         {
192             q.push_back(v[i]);
193
194             while (dq.size() != 0 && dq.getTail()->data > v[i])
195                 dq.pop_back();
196             dq.push_back(v[i]);
197         }
198     }
199
200     for (; i < n; i++)
201     {
202         sum += dq.getHead()->data;
203
204
205
206         if (q.getHead()->data == dq.getHead()->data)
207         {
208             q.pop_front();
209             dq.pop_front();
210         }
211         else
212             q.pop_front();
213
214
215
216         q.push_back(v[i]);
217
218         while (dq.size() != 0 && dq.getTail()->data > v[i])
219             dq.pop_back();
220         dq.push_back(v[i]);
221
222     }
```

```
223     }  
224  
225     sum += dq.getHead()->data; //last sub  
226  
227     cout << sum;  
228     return 0;  
229 }  
230
```

Line: 1 Col: 1

[Upload Code as File](#)☐ Test against custom input

Run Code

Submit Code

[Contest Calendar](#) | [Interview Prep](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)