**CÂU 01: CÁNH TAY ROBOT**

- Cách 1 (3.5đ): Đề bảo gì làm đó, cứ lặp lại khi nào chưa lấy hết sản phẩm thì tăng lên một lượt: lấy từ trái sang phải. Tham khảo code: <https://pastebin.com/rkfF8L9s>

- Cách 2 (5đ): Thấy rằng số 2 nếu nằm sau số 1 thì lấy cùng lượt với số 1 còn không thì phải lấy lượt khác. Các số khác tương tự. Nên ta sắp xếp mảng a này lại kèm theo chỉ số. Khi đó nếu số x có vị trí lớn hơn (x-1) thì lấy cùng lượt với x. Tham khảo code: <https://pastebin.com/230HVEsD>

- Ý tưởng bài này chú trọng vào phần sắp xếp và in mảng, tuy nhiên nhiều bạn gặp khó khăn ở bài này.

**CÂU 02: CẶP NGUYÊN TỐ TƯƠNG ĐƯƠNG**

- Cách 1 (50%): Phân tích hết các số từ L đến R ra thừa số nguyên tố, rồi tính tích các ước số nguyên tố này, lưu lại. Sau đó thì duyệt cặp (i,j) tương đương nếu có cùng tập ước nguyên tố. Tham khảo code sau phân tích ra thừa số nguyên tố bằng duyệt căn: <https://pastebin.com/fu4NMqwp>

- Cách 2 (100%): Cũng phân tích hết ra thừa số nguyên tố nhưng phân tích bằng cách sử dụng Eratos. Dùng mảng đánh dấu cnt[x] là số lượng các số có tập ước nguyên tố là x. Khi duyệt i thì ta không cần phải duyệt j nữa mà chỉ cần dựa vào mảng cnt này để đếm. Tham khảo: <https://pastebin.com/8MczueRj>

- Bài này liên quan đến kiến thức phân tích thừa số nguyên tố, đây là kiến thức khá cơ bản mà các em cần nắm để tham gia các kỳ thi HSG. Bài MATONG năm ngoái cũng có thể giải bằng cách phân tích ra thừa số nguyên tố rồi đếm ước.

**CÂU 03: ĐẾM XÂU**

- Cách 1 (60%): Duyệt hai for i và j hết các xâu con, xâu nào thoả mãn điều kiện thì đếm. Tham khảo code vét hai for: <https://pastebin.com/iEqKDW2d>

-  Cách 2 (100%): Sử dụng kiến thức mảng cộng dồn (Prefix sum) đoạn [u..v] có tổng bằng 0 khi sum[v] – sum[u-1] = 0 <=> sum[v] = sum[u-1]. Từ đây khi duyệt đến i có tổng cộng dồn là sum, khi đó ta chỉ quan tâm có bao nhiêu kí tự đã duyệt qua giống với i và có mảng cộng dồn là sum. Để đánh dấu lại số lượng ta dùng 26 map cho 26 loại kí tự. Tham khảo code: <https://pastebin.com/Za1pG2G5>

**CÂU 04: CÂY TRE TRĂM NGÀN ĐỐT**

- Ý tưởng bài này dùng QHĐ để giải. Gọi dp[i] là số đoạn tách được ít nhất nếu tách đoạn a[1]..a[i]. Khi đó ta có dp[i] = min(dp[j-1] + 1) nếu các đốt tre a[j]..a[i] có thể tách ra làm một đoạn. Để dp[i] nhỏ nhất thì ta lấy j càng bé càng tốt.

- Cách 1 (60%): Tìm j bằng cách duyệt vòng for. Tham khảo: <https://pastebin.com/sicuBqVp>

- Cách 2 (80%): Tìm j nhanh trong log(n) bằng multiset hoặc những cấu trúc tương đương. Tham khảo code: <https://pastebin.com/fmXiFvPU>

- Cách 3 (100%): Tìm j nhanh bằng kỹ thuật hai con trỏ và deque. Đầu không định đưa subtask này vào đề nhưng sau cùng đưa vào để các bạn lấy điểm 20.

- Tham khảo code: <https://pastebin.com/mmUc6BB3>

- Bạn nào vét tốt 4 bài thì được 12đ khả năng đạt giải rất cao. Còn vét mà làm tốt sắp xếp bài 1 thì đã đạt 13.5đ rơi vào tốp giải ba hoặc nhì. Nếu nắm chắc kỹ thuật phân tích thừa số nguyên tố thì đạt tầm 16đ là giải nhì chắc chắn.

- Chương trình VD HSG năm nay tạm kết thúc, hẹn gặp lại các bạn năm sau. Trong tuần sau có thể sẽ có chương trình replay đề 2020-2021 của các tỉnh. Nếu có sẽ thi online ngay trên hệ thống vdcoder. Các bạn xem phần contest để biết thông tin.

Arobot

Sub1:

1. #include <bits/stdc++.h>
2. using namespace std;
3. const int N = 4e5 + 5;
4. int n, m, a[N];
6. int main()
7. {
8. freopen("AROBOT.inp", "r", stdin);
9. freopen("AROBOT.out", "w", stdout);
10. ios::sync\_with\_stdio(0), cin.tie(0), cout.tie(0);
11. cin >> n;
12. for (int i = 1; i <= n; i++)
13. cin >> a[i];
14. m = 1;
15. while (m <= n)
16. {
17. for (int i = 1; i <= n; i++)
18. if (a[i] == m)
19. {
20. cout << a[i] << " ";
21. m++;
22. }
23. cout << "**\n**";
24. }
25. return 0;
26. }

Sub2

1. #include <bits/stdc++.h>
2. using namespace std;
3. const int N = 4e5 + 5;
4. int n;
5. pair<int, int> a[N];
7. int main()
8. {
9. freopen("AROBOT.inp", "r", stdin);
10. freopen("AROBOT.out", "w", stdout);
11. ios::sync\_with\_stdio(0), cin.tie(0), cout.tie(0);
12. cin >> n;
13. for (int i = 1; i <= n; i++)
14. {
15. cin >> a[i].first;
16. a[i].second = i;
17. }
18. sort(a + 1, a + 1 + n);
19. cout << a[1].first;
20. for (int i = 2; i <= n; i++)
21. if (a[i].second > a[i - 1].second)
22. cout << " " << a[i].first;
23. else
24. cout << "**\n**"
25. << a[i].first;
26. return 0;
27. }

Bgprime

1. #include <bits/stdc++.h>
2. using namespace std;
3. const int N = 2e6 + 5;
4. int L, R, e[N + 5], cnt[N + 5];
6. int main()
7. {
8. freopen("BGPRIME.inp", "r", stdin);
9. freopen("BGPRIME.out", "w", stdout);
10. ios::sync\_with\_stdio(0), cin.tie(0), cout.tie(0);
11. //eratos cai tien: e[i] = i neu i la SNT
12. //e[j] = i neu j nhan i la thua so nguyen to be nhat
13. for (int i = 1; i <= N; i++)
14. e[i] = i;
15. for (int i = 2; i \* i <= N; i++)
16. if (e[i] == i)
17. for (int j = i \* i; j <= N; j += i)
18. if (e[j] == j)
19. e[j] = i;
20. //De bao gi lam do: Tim tap uoc nguyen to moi so
21. //bang cach phan tich ra thua so nguyen to
22. cin >> L >> R;
23. fill(cnt, cnt + N, 0);
24. long long ans = 0;
25. for (int i = L; i <= R; i++)
26. {
27. int x = i, uocNT = 1, u;
28. while (x > 1)
29. {
30. uocNT \*= e[x], u = e[x];
31. while (x % u == 0)
32. x /= u;
33. }
34. ans += cnt[uocNT];
35. cnt[uocNT]++;
36. }
37. cout << ans;
38. return 0;
39. }

Cntstr

1. #include <bits/stdc++.h>
2. using namespace std;
3. const int N = 1e5 + 5;
4. long long n, c[26];
5. map<long long, int> sg[26];
6. string s;
8. int main()
9. {
10. freopen("CNTSTR.inp", "r", stdin);
11. freopen("CNTSTR.out", "w", stdout);
12. ios::sync\_with\_stdio(0), cin.tie(0), cout.tie(0);
13. for (int i = 0; i < 26; i++)
14. cin >> c[i];
15. cin >> s;
16. long long ans = 0, sum = 0;
17. for (int i = 0; i < s.size(); i++)
18. {
19. int u = s[i] - 'a';
20. sum += c[u];
21. ans += sg[u][sum - c[u]];
22. sg[u][sum]++;
23. }
24. cout << ans;
25. return 0;
26. }
27. #include <bits/stdc++.h>
28. using namespace std;
29. const int N = 1e5 + 5;
30. long long n, c[26];
31. map<long long, int> sg[26];
32. string s;
34. int main()
35. {
36. freopen("CNTSTR.inp", "r", stdin);
37. freopen("CNTSTR.out", "w", stdout);
38. ios::sync\_with\_stdio(0), cin.tie(0), cout.tie(0);
39. for (int i = 0; i < 26; i++)
40. cin >> c[i];
41. cin >> s;
42. long long ans = 0, sum = 0;
43. for (int i = 0; i < s.size(); i++)
44. {
45. int u = s[i] - 'a';
46. sum += c[u];
47. ans += sg[u][sum - c[u]];
48. sg[u][sum]++;
49. }
50. cout << ans;
51. return 0;
52. }
53. #include <bits/stdc++.h>
54. using namespace std;
55. const int N = 1e5 + 5;
56. int n, k, t, dp[N], a[N];
58. int main()
59. {
60. freopen("POACEAE.inp", "r", stdin);
61. freopen("POACEAE.out", "w", stdout);
62. ios::sync\_with\_stdio(0), cin.tie(0), cout.tie(0);
63. int gt;
64. cin >> gt;
65. while (gt--)
66. {
67. cin >> n >> k >> t;
68. for (int i = 1; i <= n; i++)
69. cin >> a[i];
70. deque<int> dq1, dq2;
71. fill(dp, dp + N, -1), dp[0] = 0;
72. int j = 0;
73. for (int i = 1; i <= n; i++)
74. {
75. ///duy tri min max
76. while (dq1.size() && a[dq1.back()] > a[i])
77. dq1.pop\_back();
78. while (dq2.size() && a[dq2.back()] < a[i])
79. dq2.pop\_back();
80. dq1.push\_back(i), dq2.push\_back(i);
81. ///cap nhat lai doan [j..i] neu doan ko thoa man
82. while (j < i && (a[dq2.front()] - a[dq1.front()] > t || dp[j] == -1))
83. {
84. j++;
85. while (dq1.size() && dq1.front() <= j)
86. dq1.pop\_front();
87. while (dq1.size() && dq2.front() <= j)
88. dq2.pop\_front();
89. }
90. //Neu tim thay j nho nhat thoa man thi cap nhat dp[i]
91. if (i - j >= k && dp[j] > -1)
92. dp[i] = dp[j] + 1;
93. //cout << i << " " << j << " " << dp[j] << "\n";
94. }
95. cout << dp[n] << "**\n**";
96. }
97. return 0;
98. }
99. #include <bits/stdc++.h>
100. using namespace std;
101. const int N = 1e5 + 5;
102. int n, k, t, dp[N], a[N];
104. int main()
105. {
106. freopen("POACEAE.inp", "r", stdin);
107. freopen("POACEAE.out", "w", stdout);
108. ios::sync\_with\_stdio(0), cin.tie(0), cout.tie(0);
109. int gt;
110. cin >> gt;
111. while (gt--)
112. {
113. cin >> n >> k >> t;
114. for (int i = 1; i <= n; i++)
115. cin >> a[i];
116. fill(dp, dp + N, -1), dp[0] = 0;
117. for (int i = 1; i <= n; i++)
118. {
119. int amin = a[i], amax = a[i];
120. for (int j = i; j >= 1; j--)
121. {
122. amin = min(amin, a[j]);
123. amax = max(amax, a[j]);
124. if (i - j + 1 >= k && amax - amin <= t && dp[j - 1] > -1)
125. dp[i] = dp[j - 1] + 1;
126. }
127. }
128. cout << dp[n] << "**\n**";
129. }
130. return 0;
131. }