

Iftala Zahri Sukmana


5025221002


Struktur Data (D)


Praktikum 4


Rekap:


1. **_R / Rook** : **G DULU**
2. **DAYAT9 / Dayat Terkejut** : **CORRECT**
3. **ERSY4 / Jumat Ersya** : **CORRECT**
4. **PPA / Perang Parit Air** : **CORRECT**


 Home


 Problemset


 Scoreboard

 Submit

 Logout

 Change password

 contest over

RANK	TEAM	SCORE	<div><div>_R</div></div>	<div><div>DAYAT9</div></div>	<div><div>ERSY4</div></div>	<div><div>PPA</div></div>
?	<div><div></div><div>Iftala Zahri Sukmana</div><div>Institut Teknologi Sepuluh Nopember</div></div>	3 6311	<div><div>5755</div><div>8 hrs</div></div>	<div><div>208</div><div>1 hr</div></div>	<div><div>208</div><div>1 hr</div></div>	

Submissions

time	problem	lang	result	
Tue, 23 May 18:56 PM	<div><div>DAYAT9</div></div>	CPP	CORRECT	<div><div></div></div>
Tue, 23 May 18:51 PM	<div><div>DAYAT9</div></div>	CPP	WRONG-ANSWER	<div><div></div></div>
Mon, 22 May 22:48 PM	<div><div>DAYAT9</div></div>	CPP	WRONG-ANSWER	<div><div></div></div>
Mon, 22 May 22:33 PM	<div><div>DAYAT9</div></div>	CPP	WRONG-ANSWER	<div><div></div></div>
Mon, 22 May 22:28 PM	<div><div>DAYAT9</div></div>	CPP	WRONG-ANSWER	<div><div></div></div>
Mon, 22 May 22:25 PM	<div><div>DAYAT9</div></div>	CPP	WRONG-ANSWER	<div><div></div></div>
Mon, 22 May 22:21 PM	<div><div>DAYAT9</div></div>	CPP	TIMELIMIT	<div><div></div></div>
Sat, 20 May 21:52 PM	<div><div>DAYAT9</div></div>	CPP	WRONG-ANSWER	<div><div></div></div>
Fri, 19 May 22:28 PM	<div><div>PPA</div></div>	CPP	CORRECT	<div><div></div></div>
Fri, 19 May 22:28 PM	<div><div>ERSY4</div></div>	CPP	CORRECT	<div><div></div></div>

Clarifications

No clarifications.

Clarification Requests

No clarification request.

request clarification

1. DAYAT7 / Dayat Terkejut

Link Ideone : <https://ideone.com/60oEFu>

Penjelasan :









Menentukan banyaknya pieces di sekitar titik start

Solusi :

Soal sendiri meminta praktikan untuk menentukan jumlahnya pieces yang bisa didapat. Untuk mendapatkannya, bisa menggunakan algoritma BFS dengan pendekatan sebagai berikut:

- Melakukan deklarasi array `a[]` dan `b[]` sebagai penentuan koordinat pada algoritma BFS. Konten dari array tersebut merepresentasikan empat kemungkinan dari pergerakan titik (atas, bawah, kiri, kanan).
- Function BFS kemudian dideklarasikan dengan nama `x` dan `y`, yang digunakan sebagai titik mulainya algoritma. Disini, saya menggunakan queue dan pair untuk menyimpan titik-titiknya.
- Tandai titik awal menjadi true, dan tambahkan variabel 'pieces', yang akan menghitung banyaknya '#' yang ditemui dalam pencarian.
- Elemen yang ada di depan queue akan dipop, dan ditambahkan ke dalam `x` dan `y` untuk dijadikan koordinat. Setelah itu, lakukan iterasi sebanyak 4 kali (pergerakan yang mungkin, ditandai dengan $nx = x + a[i]$ dan $ny = y + b[i]$).
- Lakukan kondisi dimana ketika '#' ditemui, maka dilakukan penambahan pada variabel 'pieces', dan apabila dijumpai titik yang bukan titik awal (ditandai dengan `check[nx][ny]`), maka dimasukkan ke dalam queue untuk melakukan perulangan dalam pencarian.

Setelah dicoba submit, terdapat error pada timelimit, yang membuat saya memasukkan `ios_base::sync_with_stdio(false);` untuk optimalisasi stream input dan output agar waktu eksekusi menjadi lebih cepat.

Tue, 23 May 18:55 PM	DAYAT9	CPP	CORRECT	
Tue, 23 May 18:51 PM	DAYAT9	CPP	WRONG-ANSWER	
Mon, 22 May 22:48 PM	DAYAT9	CPP	WRONG-ANSWER	
Mon, 22 May 22:33 PM	DAYAT9	CPP	WRONG-ANSWER	
Mon, 22 May 22:28 PM	DAYAT9	CPP	WRONG-ANSWER	
Mon, 22 May 22:25 PM	DAYAT9	CPP	WRONG-ANSWER	
Mon, 22 May 22:21 PM	DAYAT9	CPP	TIMELIMIT	
Sat, 20 May 21:52 PM	DAYAT9	CPP	WRONG-ANSWER	

Memang sebaiknya jangan terlalu gegabah.

2. ERSY2 / Jumat Ersya

Link Ideone : <https://ideone.com/UjwxXi>

Penjelasan :

Melakukan penghitungan banyaknya island secara DFS

Solusi :

Soal yang diberikan cukup sederhana, meminta menghitung banyaknya 1 dalam kumpulan 0 yang tersebar secara 2D. Disini pendekatan yang dilakukan adalah penggunaan DFS, dengan detail sebagai berikut

- a. Pendefinisian array 2 dimensi dengan row dan col untuk menentukan koordinatnya. Ditambahkan juga 2D Boolean array bernama visited untuk mencatat pergerakan. Setiap pergerakan yang didatangi akan ditandai sebagai 'true'
- b. Sama seperti sebelumnya, yaitu mendeklarasikan array untuk pergerakan, yang ditandai dengan {-1, 1, 0, 0} dan {0, 0, -1, 1} untuk pergerakan koordinat ke atas, bawah, kiri, dan kanan
- c. Seluruh kemungkinan pergerakan akan diiterasi sebanyak 4 kali (jumlah anggota array). Pada setiap pergerakan, akan ditambahkan sesuai dengan nilai yang dimiliki oleh indeks saat ini.
- d. Lakukan kondisi apabila koordinat yang ditinggali saat ini bernilai satu, dan belum dikunjungi. Apabila ditemui, maka function dfs() akan dilanjutkan.
- e. Diberikan juga fungsi count(), dimana melakukan inkremen pada nilai tersebut untuk menghitung jumlah '1' yang ada pada input.

3. FPP / Perang Parit Air

Link Ideone : <https://ideone.com/UdJatm>

Penjelasan :

Menghitung banyaknya petak yang terhubung dengan algoritma BFS

Solusi :

Pada dasarnya. Algoritma akan membentuk sebuah array 2 dimensi, dan melakukan BFS pada koordinat yang diinputkan, dan menghitung banyaknya petak yang terhubung pada array. Array yang telah dibuat direpresentasikan dalam bentuk vector<bool> dan dimasukkan ke dalam visited, dengan besaran n dan m.

Value dari visited dibuat 0 terlebih dahulu, yang mana kemudian koordinat awal akan didekrement untuk menjalankan zero-based indexing. Setelah itu, diinisialisasikan queue untuk menampung koordinat awal dan menandakan petak awal telah dikunjungi.

Sama seperti sebelumnya, dideklarasikan juga pergerakannya berupa {-1, 1, 0, 0} dan {0, 0, -1, 1} untuk pergerakan koordinat ke atas, bawah, kiri, dan kanan. Kemudian

dilakukanlah BFS. Koordinat diambil dari front dari queue dan iterasikan sebanyak empat kemungkinan arah pergerakan. Hasilnya kemudian dimasukkan ke kondisi, apakah petak tersebut sudah dikunjungi dan nilainya lebih kecil dari koordinat awal.

Apabila memenuhi kondisi, maka koordinat yang telah dibuat dimasukkan ke dalam queue baru dan ditandai sebagai true, untuk menandakan bahwa petak tersebut sudah dikunjungi, serta melakukan inkremen pada variabel area, menunjukkan jumlah petak yang telah dikunjungi.