Quiz PBO C – Soal 1

Brendan Timothy Mannuel

5025221177

Trapping Rain Water

A black and blue squares

Description automatically generated

Pada soal ini kita diperlukan mencari merapa banyak air (kotak) yang tertampung diantara kolom kolom yang terbentuk dari kotak kotak. Kita akan menggunakan implementasi Stack untuk mengerjakan soal ini.

A screenshot of a computer

Description automatically generated

Setelah kita coba membaca soal lagi, kita dapat melihat agar air dapat terperangkap maka kita memerlukan 2 kolom yang terdapat minimal jarak 1 diantara 2 kolom tersebut, serta ketinggian dari air ditentukan oleh kolom yang lebih rendah. Kita akan menggunakan sample testcase 2 untuk menggambarkannya. Kita akan menyimpannya di suatu array.

6

4 2 0 3 2 5

Untuk step pertama kita akan membuat suatu stack kosong. Kemudian kita akan memasukan index nya sebagai penunjuk ke tinggi agar nanti kita dapat lebih mudah mencari jarak diantara. Angka pertama akan langsung dimasukan karena stack masih kosong.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 |  |  |  |  |  |
| 4 |  |  |  |  |  |

Kemudian untuk angka selanjutnya kita akan mengecek dengan parameter apakah stack tidak kosong dan ketinggian dari kolom di stack.top() itu lebih pendek dari ketinggian dari isi array selanjutnya. Ternyata masih belum memenuhi sehingga kita akan mengepush lagi.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 1 |  |  |  |  |
| 4 | 2 |  |  |  |  |

Kemudian kita akan melakukan hal yang sama lagi dengan mengecek parameter, ternyata masih salah sehingga kita akan mengepush lagi.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 |  |  |  |
| 4 | 2 | 0 |  |  |  |

Setelah itu kita akan mengechek lagi, dan ternyata ketinngian selanjutnya lebih tinggi maka kita akan menyimpan stack.top() sekarang dan kita akan mengepop nya.

Tingiatas = 0

Index skrg = 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 1 |  |  |  |  |
| 4 | 2 |  |  |  |  |

Kemudian kita akan melihat apakah stack telah kosong, ternyata tidak. Selanjutnya kita akan mencari jarak diantara kedua kolom dengan mengurangi index nya

jarakantara = i - st.top() – 1

3 – 1 – 1 = 1 ( jarak antara)

Kemudian kita akan mencari kolom yang lebih rendah diantara st.top() dan ketinggian dari index sekarang. Ternyata lebih rendah st.top() karena memiliki tinggi 2 sedangkan index sekarang memiliki ketinggian 3. Kemudian kita akan melakukan pengurangan dengan angka yang telah kita simpan (mengantisipasi bila didalam kolom tersebut ada balok yang dibawahnya) yaitu 0, 2 – 0 = 2. Kemudian kita akan mengalikan nya dengan jarak yaitu 2 x 1 = 2. Kemudian kita akan menyimpan hasil ini dan melanjutkan iterasi.

Ans = 2.

Kemudian kita akan melakukan pengecheckan apakah st.top() lebih kecil daripada index sekarang ternyata benar sehingga kita akan melakukan iterasi lagi

Tingiatas = 2

Index skrg = 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 |  |  |  |  |  |
| 4 |  |  |  |  |  |

Kemudian kita akan melihat apakah stack telah kosong, ternyata tidak. Selanjutnya kita akan mencari jarak diantara kedua kolom dengan mengurangi index nya

jarakantara = i - st.top() – 1

3 – 0– 1 = 2 ( jarak antara)

Kemudian kita akan mencari kolom yang lebih rendah diantara st.top() dan ketinggian dari index sekarang. Ternyata lebih rendah index sekarang karena memiliki tinggi 3 sedangkan st.top() memiliki ketinggian 4. Kemudian kita akan melakukan pengurangan dengan angka yang telah kita simpan (mengantisipasi bila didalam kolom tersebut ada balok yang dibawahnya) yaitu 0, 3 – 2 = 1. Kemudian kita akan mengalikan nya dengan jarak yaitu 1 x 2 = 2. Kemudian kita akan menyimpan hasil ini dan melanjutkan iterasi.

Ans = 2 + 2 = 4.

Kita akan melakukan pengecekan untuk index selanjutnya. Ternyata lebih rendah dari stack.top() maka kita akan mengepush nya.

Kemudian kita akan melakukan pengecheckan apakah st.top() lebih kecil daripada index sekarang ternyata salah sehingga kita akan mengepush index sekarang.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 3 |  |  |  |  |
| 4 | 3 |  |  |  |  |

Kemudian untuk angka selanjutnya kita akan mengecek dengan parameter apakah stack tidak kosong dan ketinggian dari kolom di stack.top() itu lebih pendek dari ketinggian dari isi array selanjutnya. Ternyata masih belum memenuhi sehingga kita akan mengepush lagi.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 3 | 4 |  |  |  |
| 4 | 3 | 2 |  |  |  |

Setelah itu kita akan mengechek lagi, dan ternyata ketinngian selanjutnya lebih tinggi maka kita akan menyimpan stack.top() sekarang dan kita akan mengepop nya.

Tingiatas = 2

Index skrg = 5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 3 |  |  |  |  |
| 4 | 3 |  |  |  |  |

Kemudian kita akan melihat apakah stack telah kosong, ternyata tidak. Selanjutnya kita akan mencari jarak diantara kedua kolom dengan mengurangi index nya

jarakantara = i - st.top() – 1

5 – 3 – 1 = 1 ( jarak antara)

Kemudian kita akan mencari kolom yang lebih rendah diantara st.top() dan ketinggian dari index sekarang. Ternyata lebih rendah st.top() karena memiliki tinggi 3 sedangkan index sekarang memiliki ketinggian 5. Kemudian kita akan melakukan pengurangan dengan angka yang telah kita simpan (mengantisipasi bila didalam kolom tersebut ada balok yang dibawahnya) yaitu 0, 3 – 2 = 1. Kemudian kita akan mengalikan nya dengan jarak yaitu 1 x 1 = 1. Kemudian kita akan menyimpan hasil ini dan melanjutkan iterasi.

Ans = 4 + 1 = 5.

Kemudian kita akan melakukan pengecheckan apakah st.top() lebih kecil daripada index sekarang ternyata benar sehingga kita akan melakukan iterasi lagi

Tingiatas = 3

Index skrg = 5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 |  |  |  |  |  |
| 4 |  |  |  |  |  |

Kemudian kita akan melihat apakah stack telah kosong, ternyata tidak. Selanjutnya kita akan mencari jarak diantara kedua kolom dengan mengurangi index nya

jarakantara = i - st.top() – 1

5 – 0 – 1 = 4 ( jarak antara)

Kemudian kita akan mencari kolom yang lebih rendah diantara st.top() dan ketinggian dari index sekarang. Ternyata lebih rendah st.top() karena memiliki tinggi 4 sedangkan index sekarang memiliki ketinggian 5. Kemudian kita akan melakukan pengurangan dengan angka yang telah kita simpan (mengantisipasi bila didalam kolom tersebut ada balok yang dibawahnya) yaitu 0, 4 – 3 = 1. Kemudian kita akan mengalikan nya dengan jarak yaitu 1 x 4 = 4. Kemudian kita akan menyimpan hasil ini dan melanjutkan iterasi.

Ans = 5 + 4 = 9.

Kemudian kita akan melakukan pengecheckan apakah st.top() lebih kecil daripada index sekarang ternyata benar sehingga kita akan melakukan iterasi lagi

Tingiatas = 4

Index skrg = 5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Kemudian kita akan melihat apakah stack telah kosong, ternyata iya sehingga kita akan melakukan pemutusan pada loop dan mengepush index sekarang

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 5 |  |  |  |  |  |
| 5 |  |  |  |  |  |

Karena i sudah memiliki nilai yang sama dengan m maka kita tinggal melakukan output ans yaitu 9.

A screen shot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

Quiz PBO C – Soal 2

Brendan Timothy Mannuel

5025221177

Arrange

*There are m students in a classroom, out of which n of them are girls. The teacher, Mr X wants to arrange all the students in a line. X thinks that his class girls are very talkative. So he doesn't want any two girls to be together. Mr X wants to know the number of ways he can arrange these m students. Help him out.*

Pada soal ini kita diperlukan mencari berapa banyak kombinasi yang dapat dibentuk dari suatu urutan tempat duduk antara cewek dan cowok dimana tidak boleh ada 2 cewek yang duduk bersebelahan. Untuk menyelesaikan permasalahan ini kita dapat menggunakan rumus kombinasi pada umumnya, kita misalkan jumlah laki laki adalah variable L, yaitu untuk laki laki maka banyak cara untuk laki laki adalah factorial dari jumlah laki laki (L!) sedangkan untuk Perempuan dapat duduk sebanyak laki-laki + 1 faktorial (L+1!) dengan kombinasi sebanyak L+1Pn . Di soal juga dijelaskan bahwa kita harus memodulo jawaban dengan  109 + 7.

A computer screen with text and images

Description automatically generated

Ternyata setelah kita coba untuk submit jawaban, ternyata masih Run Time Error

A screenshot of a computer

Description automatically generated

Setelah kita coba untuk mencari tahu kesalahan, terdapat beberapa kemungkinan kesalahan yang bisa terjadi yaitu operasi factorial memakan waktu yang terlalu lama, serta ada kondisi pada percabangan yang kurang benar.

Oleh karena itu kita dapat mencoba mengubah dengan cara menyimpan operasi factorial sekali saja saat kita menjalankan program sehingga akan mempercepat perhitungan.

A black background with colorful numbers

Description automatically generated

Selain itu kita juga akan menambahkan 1 kasus percabangan Ketika jumlah laki – berjumlah lebih sedikit dan memiliki selisih lebih dari 2 dimana pada kasus ini dipastikan ada 2 cewek yang akan selalu duduk bersebelahan sehingga kita dapat langsung mengasumsikan jawaban = 0

A computer code on a black background

Description automatically generated

Untuk mempercepat perhitungan factorial kita dapat menggunakan Fermat Little Theorem untuk mempercepat mencari kombinasi dengan memanfaatkan Modular Multiplicative Inverse, dan juga menghilangkan operasi pembagian yang juga memakan banyak cost

A computer screen shot of white and blue text

Description automatically generatedA black background with white text

Description automatically generated

Dengan memperbaiki tiga hal diatas maka kita dapat lebih mengoptimisasi kode sehingga tidak terjadi RTE.

A screenshot of a computer

Description automatically generated