

Name - Shradha Sharma

Father's name - Sanjeev Sharma

Course - BSC IT - 2 - MID TERM PRACTICAL

University Roll no - ~~2100~~ ~~2013100~~ 2023100

Student ID - 20132019

Subject - OPERATING SYSTEM

①

Q1) #include <stdio.h>

unsigned int Heap [100001], Index [100001], position [100001],
Size = 0;

unsigned int temp [100001], Temp1 [100001];

unsigned int Arr-Time [100001], cook-Time [100001], Num;

void merge (int low, int Mid, int High)

{

int i = low, j = Mid + 1, k = 0;

while (i <= Mid & j <= High)

{

if (Arr-Time [i] <= Arr-Time [j])

{

Temp [k] = Arr-Time [i];

Temp1 [k] = cook-Time [i];

i++;

k++;

}

else

{

Temp [k] = Arr-Time [j];

Temp1 [k] = ~~Arr~~ cook-Time [j];

j++;

k++;

}

}

if (i <= Mid)

{

int I;

for (I = i; I <= Mid; I++)

Shradha.

(2)

```

{ Temp[k] = Arr_Time[I];   Temp1[k] = Cook_Time[I]; k++; }
}
else if (j <= High)
{
    int I;
    for (I = j; I <= High; I++)
    { Temp[k] = Arr_Time[I];   Temp1[k] = Cook_Time[I]; k++; }
}
k = 0;
for (i = low; i <= High; i++)
{
    Arr_Time[i] = Temp[k];
    Cook_Time[i] = Temp1[k];
    k++;
}
}

void divide(int low, int High)
{
    if (low < High)
    {
        int Mid = (low + High) / 2;
        divide(low, Mid);
        divide(Mid + 1, High);
        merge(low, Mid, High);
    }
}

void Insert(int Node, unsigned int value)
{
    int s;
    if (position[Node] == 0)
    {
        Heap[++Size] = value;
        Index[Size] = Node;
        position[Node] = Size;
        S = Size;
    }
}

```

Shradha

③

```
else
{
    Heap [position [Node]] = value;
    s = position [Node];
}
while (s != 1)
{
    if (Heap [s/2] > Heap [s])
    {
        int t = Heap [s/2];
        Heap [s/2] = Heap [s];
        Heap [s] = t;

        t = Index [s/2];
        Index [s/2] = Index [s];
        Index [s] = t;

        position [Index [s/2]] = s/2;
        position [Index [s]] = s;
    }
    else
        break;
    s = s/2;
}

int Extract-Min()
{
    int N = Index [1];
    int s = 1;

    // printf ("%d\n", Heap [1]);
    position [N] = ~1;
    Index [1] = Index [size];
    position [Index [size]] = 1;
    Heap [1] = Heap [size - 1];
    while (1)
```

Shradha

```

1 int T;
if (Heap[s*2] < Heap[s] && s*2 <= size || Heap[s*2+1] < Heap[s]
    && s*2+1 <= size)
{
    if (Heap[s*2] < Heap[s*2+1])
        T = s*2;
    else
        T = s*2+1;

    int t = Heap[T];
    Heap[T] = Heap[s];
    Heap[s] = t;

    t = Index[T];
    Index[T] = Index[s];
    Index[s] = t;

    position[Index[T]] = T;
    position[Index[s]] = s;
}
else
    break;
s = T;
}
return N;
}

void Init(int N)
{
    int i;
    for (i = 1; i <= N; i++)
    {
        position[i] = 0;
        Index[i] = 0;
        Heap[i] = 1000000000;
    }
    size = N;
}

```

Shradha

(5)

```

int main()
{
    int A-T, C-T, i=1;
    long long wait-Time=0, Time=0;
    scanf ("%d", &Num);
    //init (N);
    for (i=0; i<Num; i++)
        scanf ("%u %u", &Arr-Time [i], &cook-Time [i]);
    divide (0, Num-1);
    for (i=Num; i>=1; i--)
    {
        Arr-Time [i] = Arr-Time [i-1];
        cook-Time [i] = cook-Time [i-1];
        // printf ("%u %u\n", Arr-Time [i], cook-Time [i]);
    }
    Insert (1, cook-Time [1]);
    i=2;
    while (i<=Num && Arr-Time [i] == Arr-Time [1])
    {
        insert (i, cook-Time [i]);
        i++;
    }
    while (size != 0)
    {
        int I = Extract-Min (1);
        if (Time > Arr-Time [I])
        {
            wait-Time += Time - Arr-Time [I] + cook-Time [I];
            Time += cook-Time [I];
            // printf ("%d %d %d\n", I, Time, wait-Time);
        }
        else
        {
            Time = Arr-Time [I] + cook-Time [I];
        }
    }
}

```

Shradha.

(6)

```

wait_time += cook_time[I];
}
// printf("%d %d %d\n", I, time, wait_time);
I = 1;
while (i <= Num && Arr_time[i] <= time)
{
    Insert(i, cook_time[i]);
    i++;
}
if (I == 1 && i <= Num) // No job is before cur_time.
{
    Insert(i, cook_time[i]);
    i++;
    while (i <= Num && Arr_time[i] == Arr_time[I])
    {
        Insert(i, cook_time[i]);
        i++;
    }
}
wait_time = wait_time / Num;
printf("%d", wait_time);
// system("pause");
return 0;
}

```

Output;

3

0 3

1 9

2 5

8

Shradha.

OnlineGDB beta
online compiler and debugger for c/c++
code, compile, run, debug, share.

IDE
My Projects
Classroom **new**
Learn Programming
Programming Questions
We are Hiring
Sign Up
Login

f t + 56.1K

About • FAQ • Blog • Terms of Use • Contact Us
• GDB Tutorial • Credits • Privacy
© 2016 - 2021 GDB Online

onlinegdb.com/online_c_compiler

Run Debug Stop Share Save Beautify

Language C

main.c

```
191 {  
192     Insert(i,Cook_Time[i]);  
193     i++;  
194 }  
195 if(I==i&&i<=Num)//No job is before curr_time  
196 {  
197     Insert(i,Cook_Time[i]);  
198  
199     i++;  
200 while(i<=Num&&Arr_Time[i]==Arr_Time[I])  
201 {  
202     Insert(i,Cook_Time[i]);  
203     i++;  
204 }  
205 }  
206 }  
207 Wait_Time=Wait_Time/Num;  
208 printf("%lld",Wait_Time);  
209 // system("pause");
```

Input

```
3  
0 3  
1 9  
2 5  
8  
...Program finished with exit code 0  
Press ENTER to exit console.
```

Type here to search

29°C 10:30 22-06-2021