Name: Shubhangi Agrawal (20BCE1161)
Date: 13/4/22

Lab 11 - Dynamic memory allocation

First fit:

```
#include<bits/stdc++.h>
using namespace std;
using ll=long long;
void ynans(bool x){if(x) cout<<"YES";else cout<<"NO";}
#define vi vector<int>
#define rep(i,k,n) for(ll i=k;i<n;i++)
#define rof(i,k,n) for(ll i=k;i>n;i--)
#define pb(x) push_back(x)
#define sp(x,y) fixed<<setprecision(y)<<x
int sum() { return 0; }
template<typename T, typename... Args>
T sum(T a, Args... args) { return a + sum(args...); }
#define vi vector<int>
#define vc vector<char>
#define vs vector<string>
#define vll vector<ll>
#define vvi vector < vi >
#define pll pair<ll, ll>
#define ff first
#define ss second
#define casePrint(x,y) cout<<"Case #"<<x<<": "<<y;
#define all(c) c.begin(),c.end()
int main(){
   ll pn;
   cout<<"no. of process? ";
   cin>>pn;
   ll p[pn];
   cout<<"processes' sizes? \n";
   rep(i,0,pn){
       cout<<"P"<<i<<" ";
       cin>>p[i];
   }
   ll hn;
   cout<<"no. of holes? ";
   cin>>hn;
   ll h[hn];
```

```
   cout<<"holes' sizes? \n";
   rep(i,0,hn){
       cout<<"H"<<i<<" ";
       cin>>h[i];
   }
   cout<<"Allocated blocks/holes to processes:\n";
   ll pleft[pn],x=0;
   rep(i,0,pn){
       ll flag=0;
       rep(j,0,hn){
           if(p[i]<=h[j]){
               cout<<"P"<<i<<" -> "<<"H"<<j<<" ("<<p[i]<<")\n";
                h[j]-=p[i];
                flag=1;
                break;
           }
       }
       if(!flag){
           pleft[x++]=i;

       }
   }
   cout<<"\nLeft processes\n";
   ll f=0;
   rep(i,0,x){
       cout<<"P"<<pleft[i]<<" ";
       f=1;
   }
   if(!f)
   cout<<"\nNo process left\n";
   cout<<"Left blocks\n";
 f=0;
rep(i,0,hn){
    if(h[i]!=0){
        cout<<"H"<<i<<": "<<h[i]<<"\t";
        f=1;
    }
}
if(!f)
cout<<"\nNo blocks left\n";
```

```
}
```

Output:

```
PS E:\VIT\4thsem\OS\lab\linuxpractice\20bce1161\lab11> cd "e
20bce1161\lab11\" ; if ($?) { g++ first.cpp -o first } ; if
no. of process? 4
processes' sizes?
P0 212
P1 417
P2 112
P3 426
no. of holes? 5
holes' sizes?
H0 100
H1 500
H2 200
H3 300
H4 600
Allocated blocks/holes to processes:
P0 -> H1 (212)
P1 -> H4 (417)
P2 -> H1 (112)

Left processes
P3
Left blocks
H0: 100 H1: 176 H2: 200 H3: 300 H4: 183
```

```
PS E:\VIT\4thsem\OS\lab\linuxpractice\20bce1161\lab11> cd "e:\v
no. of process? 4
processes' sizes?
P0 100
P1 200
P2 300
P3 400
no. of holes? 5
holes' sizes?
H0 120
H1 230
H2 310
H3 420
H4 500
Allocated blocks/holes to processes:
P0 -> H0 (100)
P1 -> H1 (200)
P2 -> H2 (300)
P3 -> H3 (400)

Left processes

No process left
Left blocks
H0: 20  H1: 30  H2: 10  H3: 20  H4: 500
```

Worst fit:

```
#include<bits/stdc++.h>
using namespace std;
using ll=long long;
void ynans(bool x){if(x) cout<<"YES";else cout<<"NO";}
#define vi vector<int>
#define rep(i,k,n) for(ll i=k;i<n;i++)
#define rof(i,k,n) for(ll i=k;i>n;i--)
#define pb(x) push_back(x)
#define sp(x,y) fixed<<setprecision(y)<<x
int sum() { return 0; }
template<typename T, typename... Args>
T sum(T a, Args... args) { return a + sum(args...); }
#define vi vector<int>
#define vc vector<char>
#define vs vector<string>
```

```
#define vll vector<ll>
#define vvi vector < vi >
#define pll pair<ll, ll>
#define ff first
#define ss second
#define casePrint(x,y) cout<<"Case #"<<x<<": "<<y;
#define all(c) c.begin(),c.end()
class hole{
    public:
    int n;
    int size;
};
class process{
    public:
    int n;
    int size;
};
bool cmh(hole p1,hole p2){
    return (p1.size>p2.size);
}
int main(){
   ll pn;
   cout<<"no. of process? ";
   cin>>pn;
    process p[pn];
   cout<<"processes' sizes? \n";
   rep(i,0,pn){
       cout<<"P"<<i<<" ";
        p[i].n=i;
       cin>>p[i].size;
   }
   ll hn;
   cout<<"no. of holes? ";
   cin>>hn;
   hole h[hn];
   cout<<"holes' sizes? \n";
   rep(i,0,hn){
       cout<<"H"<<i<<" ";
       h[i].n=i;
       cin>>h[i].size;
   }
   cout<<"Allocated blocks/holes to processes:\n";
   ll pleft[pn],x=0;
```

```
   rep(i,0,pn){
       ll flag=0;
       sort(h,h+hn,cmh);
       rep(j,0,hn){
           if(p[i].size<=h[j].size){
               cout<<"P"<<i<<" -> "<<"H"<<h[j].n<<"
("<<p[i].size<<")\n";
                h[j].size-=p[i].size;
                flag=1;
                break;

           }
       }
       if(!flag){
           pleft[x++]=i;

       }
   }
    cout<<"\nLeft processes\n";
   ll f=0;
   rep(i,0,x){
       cout<<"P"<<pleft[i]<<" ";
       f=1;
   }
   if(!f)
   cout<<"\nNo process left\n";
   cout<<"\nLeft blocks\n";
    f=0;
    rep(i,0,hn){
        if(h[i].size!=0){
            cout<<"H"<<h[i].n<<": "<<h[i].size<<"\t";
            f=1;
        }
    }
    if(!f)
    cout<<"\nNo blocks left\n";


}
```

Output:

```
PS E:\VIT\4thsem\OS\lab\linuxpractice\20bce1161\lab11> cd "e:\VIT\4thsem
no. of process? 4
processes' sizes?
P0 100
P1 200
P2 300
P3 400
no. of holes? 5
holes' sizes?
H0 120
H1 230
H2 310
H3 420
H4 500
Allocated blocks/holes to processes:
P0 -> H4 (100)
P1 -> H3 (200)
P2 -> H4 (300)

Left processes
P3
Left blocks
H2: 310 H1: 230 H3: 220 H0: 120 H4: 100
```

```
PS E:\VIT\4thsem\OS\lab\linuxpractice\20bce1161\lab11> cd "e:\VI
20bce1161\lab11\" ; if ($?) { g++ worst.cpp -o worst } ; if ($?)
no. of process? 4
processes' sizes?
P0 212
P1 417
P2 112
P3 426
no. of holes? 5
holes' sizes?
H0 100
H1 500
H2 200
H3 300
H4 600
Allocated blocks/holes to processes:
P0 -> H4 (212)
P1 -> H1 (417)
P2 -> H4 (112)

Left processes
P3
Left blocks
H3: 300 H4: 276 H2: 200 H0: 100 H1: 83
```

Best fit:

```cpp
#include<bits/stdc++.h>
using namespace std;
using ll=long long;
void ynans(bool x){if(x) cout<<"YES";else cout<<"NO";}
#define vi vector<int>
#define rep(i,k,n) for(ll i=k;i<n;i++)
#define rof(i,k,n) for(ll i=k;i>n;i--)
#define pb(x) push_back(x)
#define sp(x,y) fixed<<setprecision(y)<<x
int sum() { return 0; }
template<typename T, typename... Args>
T sum(T a, Args... args) { return a + sum(args...); }
#define vi vector<int>
#define vc vector<char>
#define vs vector<string>
```

```
#define vll vector<ll>
#define vvi vector < vi >
#define pll pair<ll, ll>
#define ff first
#define ss second
#define casePrint(x,y) cout<<"Case #"<<x<<": "<<y;
#define all(c) c.begin(),c.end()
class hole{
    public:
    int n;
    int size;
};
class process{
    public:
    int n;
    int size;
};
bool cmh(hole p1,hole p2){
    return (p1.size<p2.size);
}
int main(){
   ll pn;
   cout<<"no. of process? ";
   cin>>pn;
    process p[pn];
   cout<<"processes' sizes? \n";
   rep(i,0,pn){
       cout<<"P"<<i<<" ";
        p[i].n=i;
       cin>>p[i].size;
   }
   ll hn;
   cout<<"no. of holes? ";
   cin>>hn;
   hole h[hn];
   cout<<"holes' sizes? \n";
   rep(i,0,hn){
       cout<<"H"<<i<<" ";
       h[i].n=i;
       cin>>h[i].size;
   }
   cout<<"Allocated blocks/holes to processes:\n";
   ll pleft[pn],x=0;
```

```
   rep(i,0,pn){
       ll flag=0;
       sort(h,h+hn,cmh);
       rep(j,0,hn){
           if(p[i].size<=h[j].size){
               cout<<"P"<<i<<" -> "<<"H"<<h[j].n<<"
("<<p[i].size<<")\n";
                h[j].size-=p[i].size;
                flag=1;
                break;

           }
       }
       if(!flag){
           pleft[x++]=i;

       }
   }
    cout<<"\nLeft processes\n";
   ll f=0;
   rep(i,0,x){
       cout<<"P"<<pleft[i]<<" ";
       f=1;
   }
   if(!f)
   cout<<"\nNo process left\n";
   cout<<"\nLeft blocks\n";
    f=0;
    rep(i,0,hn){
        if(h[i].size!=0){
            cout<<"H"<<h[i].n<<": "<<h[i].size<<"\t";
            f=1;
        }
    }
    if(!f)
    cout<<"\nNo blocks left\n";


}
```

Output:

```
PS E:\VIT\4thsem\OS\lab\linuxpractice\20bce1161\lab11> cd
thsem\OS\lab\linuxpractice\20bce1161\lab11\" ; if ($?) { g
pp -o best } ; if ($?) { .\best }
no. of process? 4
processes' sizes?
P0 212
P1 417
P2 112
P3 426
no. of holes? 5
holes' sizes?
H0 100
H1 500
H2 200
H3 300
H4 600
Allocated blocks/holes to processes:
P0 -> H3 (212)
P1 -> H1 (417)
P2 -> H2 (112)
P3 -> H4 (426)

Left processes

No process left

Left blocks
H1: 83  H3: 88  H2: 88  H0: 100 H4: 174
```

```
PS E:\VIT\4thsem\OS\lab\linuxpractice\20bce1161\lab11> cd
thsem\OS\lab\linuxpractice\20bce1161\lab11\" ; if ($?) { g
pp -o best } ; if ($?) { .\best }
no. of process? 4
processes' sizes?
P0 100
P1 200
P2 300
P3 400
no. of holes? 5
holes' sizes?
H0 120
H1 230
H2 310
H3 420
H4 500
Allocated blocks/holes to processes:
P0 -> H0 (100)
P1 -> H1 (200)
P2 -> H2 (300)
P3 -> H3 (400)

Left processes

No process left

Left blocks
H2: 10  H0: 20  H1: 30  H3: 20  H4: 500
```