Operating System Practicals

Problem

Krunal Rank U18CO081

1. To implement first fit, best fit and worst fit storage allocation algorithms for memory management.

```
#include <bits/stdc++.h>
#define N 10
#define pii pair<int, pair<int,pair<int, int>>>
#define F first
#define S second
using namespace std;
   int start;
  bool occupied = false;
  Node *next;
  Node *prev;
  Node (int s, int e, bool o = false, int pid = -1, Node *n = NULL, Node *p = NULL)
       this->start = s;
       this->end = e;
       this->occupied = o;
       this->prev = p;
       this->pid = pid;
struct Process
  int pid, arrival_time, burst_time, memory_requirement;
   Process(int p, int a, int b, int m)
       this->pid = p;
       this->memory_requirement = m;
```

```
void print linked list(Node* main memory) {
  Node* head=main memory;
  cout<<"Start\t\tEnd\t\tOccupied\tPID"<<endl;</pre>
  while(head){
cout<<head->start<<"\t\t"<<head->end<<"\t\t"<<head->occupied<<"\t\t"<<head->pid<<"\t\
"<<endl;
      head = head->next;
Node* find first fit node(Node* main memory,int mem req){
  Node* head = main memory;
  while(head){
       if(head->end - head->start+1>=mem req && head->occupied==false) return head;
      head = head->next;
Node* find best fit node(Node* main memory,int mem req){
  Node* head = main memory;
  Node* req = NULL;
  while(head) {
       if(head->end - head->start+1>=mem req && head->occupied==false) {
           if(!req) req = head;
           else if(req->end-req->start>head->end-head->start) req = head;
      head = head->next;
   return req;
Node* find worst fit node(Node* main memory,int mem req){
  Node* head = main memory;
  Node* req = NULL;
  while(head) {
       if(head->end - head->start+1>=mem req && head->occupied==false){
           if(!req) req = head;
           else if(req->end-req->start<head->end-head->start) req = head;
      head = head->next;
```

```
return req;
Node* find occupied node(Node* main memory,int pid){
  Node* head = main memory;
  while(head){
       if(head->pid==pid) return head;
void fit algorithm(vector<Process *> &processes,int arg = 0)
  Node *main memory = new Node (0, 19);
  if(arg==0)
   else if(arg==1)
       cout<<"Best Fit Memory Allocation Algorithm"<<endl;</pre>
   else if(arq==2)
  priority queue<pii, vector<pii>, greater<pii>> pq;
   for(auto i: processes)
pq.push({i->arrival time,{1,{i->memory requirement,i->pid}}});
   set<int> free times;
  while(!pq.empty()){
       pii top = pq.top();
       pq.pop();
       int a_time = top.F;
       int is entry = top.S.F;
       int mem req = top.S.S.F;
       int pid = top.S.S.S;
       Process* p = processes[pid];
cout<<"-----
1;
       cout<<"Current Process at Queue Head: "<<p->pid<<endl;</pre>
       if(is entry){
           Node* result;
           if(arg==0) result = find first fit node(main memory, mem req);
           else if(arg==1) result = find best fit node(main memory, mem req);
```

```
else if(arg==2) result = find worst fit node(main memory, mem req);
if(!result){
    pii t1 = pq.top();
    pq.push({*free times.begin(), {1, {mem req,pid}}});
    cout<<"Not Slot Found!"<<endl;</pre>
    int end = result->end;
    Node* next = result->next;
    result->occupied = true;
    result->pid = p->pid;
    result->end = result->start + mem req - 1;
    int start = result->end + 1;
    Node* new node = new Node(start, end);
    result->next = new node;
    new node->prev = result;
    if(next && next->occupied==false) {
        new node->end = next->end;
        new node->next = next->next;
        if(next->next)
            next->next->prev = new node;
    }else{
        new node->next = next;
        if(next)
            next->prev = new node;
    pq.push({a time+p->burst time, {0, {mem req,pid}}});
    free times.insert(a time+p->burst time);
    cout<<"Slot Found!"<<endl;</pre>
free times.erase(a time);
Node* occupied node = find occupied node(main memory,p->pid);
Node* prev node = occupied node->prev;
Node* next node = occupied node->next;
if(!prev node || prev node->occupied==true){
    if(next node->occupied==false) {
        occupied node->end = next node->end;
        occupied node->next = next node->next;
        if(next node->next) next node->next->prev = occupied node;
        occupied node->occupied = false;
        occupied node->pid = -1;
    }else{
```

```
occupied node->pid = -1;
                   cout<<"Removed Node! No Merging!"<<endl;</pre>
           else if(!next node || next node->occupied==true){
               if(prev node->occupied==false) {
                   prev node->end = occupied node->end;
                   prev node->next = occupied node->next;
                   if(occupied node->next) occupied node->next->prev = prev node;
                   free(occupied node);
                   cout<<"Removed Node! Merged with Prev Node!"<<endl;</pre>
               }else{
                   occupied node->occupied = false;
                   occupied node->pid = -1;
                   cout<<"Removed Node! No Merging!"<<endl;</pre>
               prev node->end = next node->end;
               prev node->next = next node->next;
               if(next node->next) next node->next->prev = prev node;
               free(next node);
               free(occupied node);
               cout<<"Removed Node! Merged with Prev Node and Next Node!"<<endl;</pre>
       cout<<"Main Memory Status"<<endl;</pre>
       print linked list(main memory);
nt main()
  srand(time(NULL));
  vector<Process *> processes;
       processes.push_back(new Process(i, rand() % 10, rand() % 10+1, rand() %
10+1));
```

occupied node->occupied = false;

```
cout<<"Processes:"<<endl;
cout<<"PID\t\tA.Time\tB.Time\tMemory Requirement"<<endl;
for(auto i: processes)
cout<<ii->pid<<"\t\t"<<i->arrival_time<<"\t\t"<<i->burst_time<<"\t\t"<<i->memory_requirement<<"\t\t"<<endl;
fit_algorithm(processes,2);
}</pre>
```

First Fit:

FIRST FIT:			
Processes:			
			Memory Requirement
	6	3	10
	5	3	1
	9	6	10
3	3	6	6
	6	5	1
	1	2	1
	1	3	1
	9	9	10
	0	2	9
	8	6	9
irst Fit	Memory Allo	cation Algor	ithm
urrent Pr	ocess at Que	eue Head: 8	
lot Found			
ain Memor	y Status		
tart	End	Occupied	PID
	8	1	8
	19	0	-1
urrent Pr	ocess at Que	eue Head: 5	
lot Found			
ain Memor	y Status		
tart	End	Occupied	PID
	8	1	8
	9	1	5
0	19	0	-1
urrent Pr	ocess at Que	eue Head: 6	
lot Found			
ain Memor	y Status		
tart	- End	Occupied	PID

0	8	1	8
9	9	1	5
10	10	1	6
11	19	0	-1
	cess at Que		
	e! No Mergin	ng!	
Main Memory			
Start	End	Occupied	PID
0	8	0	-1
9	9	1	5
10	10	1	6
11	19	0	-1
Current Pro	cess at Que	ue Head: 5	
Removed Nod	e! Merged w	ith Prev Nod	e!
Main Memory	Status		
Start	End	Occupied	PID
0	9	0	-1
10	10	1	6
11	19	0	-1
Current Pro	cess at Que	ue Head: 3	
Slot Found!			
Main Memory	Status		
Start	End	Occupied	PID
0	5	1	3
6	9	0	-1
10	10	1	6
11	19	0	-1
Current Pro	cess at Que	ue Head: 6	
Removed Nod	e! Merged wi	ith Prev Nod	e and Next Node!
Main Memory	Status		
Start	End	Occupied	PID
	5		
6	19	0	-1
Current Pro	cess at Que	ue Head: 1	
Slot Found!			
Main Memory	Status		
_			
D Car C	End	Occupied	PID
		Occupied 1	
	End 5		PID 3

	19	0	-1
Slot Foun		Queue Head: 4	
	ry Status	Occupied	חוח
)	5	Occupied 1	3
	6	1	1
5			
7 8	7	1	4
3 	19 	0	-1
Current F	rocess at	Queue Head: 0	
Slot Foun			
Main Memo	ry Status		
Start	End	Occupied	PID
)	5	1	3
5	6	1	1
7	7	1	4
3	17	1	0
	19	0	-1
18 			-1
18 Current F	rocess at	Queue Head: 1	-1
18 Current F Removed N	rocess at Iode! No Me	Queue Head: 1	-1
18 Current F Removed N Main Memo	Process at Node! No Me Pry Status	Queue Head: 1	
18 Current F Removed N Main Memo	Process at Node! No Me Ory Status End	Queue Head: 1 rging! Occupied	PID
18 Eurrent F Removed N Main Memo Start	Process at Node! No Me Pry Status	Queue Head: 1	
18 Eurrent F Removed N Main Memo Start	Process at Node! No Me Pry Status End 5	Queue Head: 1 rging! Occupied 1	PID 3 -1
18 Current F Removed N Main Memo Start O	Process at Mode! No Me Pry Status End 5 6	Queue Head: 1 rging! Occupied	PID 3
18 Current F Removed N Main Memo Start O 6	Process at Node! No Me Pry Status End 5	Queue Head: 1 rging! Occupied 1	PID 3 -1
18 Current F Removed N Main Memo Start O 6	Process at Mode! No Me Pry Status End 5 6	Queue Head: 1 rging! Occupied 1 0	PID 3 -1
L8 Current F Removed N Main Memo Start) 6 7 8	Process at Jode! No Me Pry Status End 5 6 7 17 19	Queue Head: 1 rging! Occupied 1 0 1 0	PID 3 -1 4
Current F Removed N Main Memo Start O 6 7 B L8 Current F	Process at Node! No Me Pry Status End 5 6 7 17 19 Process at	Queue Head: 1 rging! Occupied 1 0 1	PID 3 -1 4
Current F Removed N Main Memo Start D 6 7 8 18 Current F Not Slot	Process at Jode! No Me Pry Status End 5 6 7 17 19 Process at Found!	Queue Head: 1 rging! Occupied 1 0 1 0	PID 3 -1 4
Current F Removed N Main Memo Start) 6 7 8 18 Current F Not Slot Main Memo	Process at Tode! No Me Pry Status End 5 6 7 17 19 Process at Found!	Queue Head: 1 rging! Occupied 1 0 1 0 2 Queue Head: 9	PID 3 -1 4 0 -1
Current F Removed N Main Memo Start) 6 7 8 L8 Current F Not Slot Main Memo	Process at Jode! No Me Pry Status End 5 6 7 17 19 Process at Found! Pry Status End	Queue Head: 1 Occupied 1 0 1 1 Queue Head: 9	PID 3 -1 4 0 -1
Current F Removed N Main Memo Start) S Current F Not Slot Main Memo Start	Process at Tode! No Me Pry Status End 5 6 7 17 19 Process at Found! Pry Status End 5	Queue Head: 1 orging! Occupied 1 0 1 1 Queue Head: 9 Occupied 1	PID 3 -1 4 0 -1 PID 33
Current F Removed N Main Memo Start) 6 7 8 18 Current F Not Slot Main Memo Start)	Process at Jode! No Me Pry Status End 5 6 7 17 19 Process at Found! Pry Status End 5 6 6	Queue Head: 1 Orging! Occupied 1 0 1 1 Queue Head: 9 Occupied 1 0	PID 3 -1 4 0 -1 PID 3 -1
Current F Removed N Main Memo Start O 6 7 8 18 Current F Not Slot Main Memo Start O 6 7 7 7	Process at Tode! No Me Pry Status End 5 6 7 17 19 Process at Found! Pry Status End 5 6 7 7 7 7 7 8 7 8 7 8 7 8 7 8 7 8 7 8 7	Queue Head: 1 orging! Occupied 1 0 1 1 Queue Head: 9 Occupied 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1	PID 3 -1 4 0 -1 PID 3 -1 4 4 0 -1
Current F Removed N Main Memo Start 0 6 7 8 18 Current F Not Slot Main Memo Start 0 6 7 7 6 7 7	Process at Jode! No Me Pry Status End 5 6 7 17 19 Process at Found! Pry Status End 5 6 6	Queue Head: 1 orging! Occupied 1 0 1 1 0 Queue Head: 9 Occupied 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	PID 3 -1 4 0 -1 PID 3 -1

Main Memory Status

Start	End	Occupied	PID
0	6	0	-1
7	7	1	4
8	17	1	0
18	19	0	-1
Current Pro	cess at Que	ue Head: 0	
Removed Nod	le! Merged w	ith Next Noc	de!
Main Memory	Status		
Start	End	Occupied	PID
0	6	0	-1
7	7	1	4
8	19	0	-1
Current Pro	cess at Que	ue Head: 9	
Slot Found!			
Main Memory	Status		
Start	End	Occupied	PID
0	6	0	-1
7	7	1	4
8	16	1	9
17	19	0	-1
Current Pro	cess at Que	ue Head: 2	
Not Slot Fo	und!		
Main Memory	Status		
Start	End	Occupied	PID
0	6	0	-1
7	7	1	4
8	16	1	9
17	19	0	-1
Current Pro	cess at Que	ue Head: 7	
Not Slot Fo	und!		
Main Memory	Status		
Start	End	Occupied	PID
0	6	0	-1
7	7	1	4
8	16	1	9
17	19	0	-1
Current Pro	cess at Que	ue Head: 4	

Removed Node! Merged with Prev Node! Main Memory Status

Start	End	Occupied	PID
0	7	0	-1
8	16	1	9
17	19	0	-1
Current Pr	ocess at Que	eue Head: 2	
Not Slot F	ound!		
Main Memor	y Status		
Start	- End	Occupied	PID
0	7	0	-1
8	16	1	9
17	19	0	-1
Current Pr	ocess at Que	eue Head: 7	
Not Slot F	ound!		
Main Memor	y Status		
Start	End	Occupied	PID
0	7	0	-1
8	16	1	9
17	19	0	-1
Current Pr	ocess at Que	eue Head: 9	
Removed No	de! Merged v	with Prev No	de and Next Node!
Main Memor	y Status		
Start	End	Occupied	PID
0	19	0	-1
Current Pr	ocess at Que	eue Head: 2	
Slot Found			
Main Memor	y Status		
Start	End	Occupied	PID
0	9	1	2
10	19	0	-1
Current Pr	ocess at Que	eue Head: 7	
Slot Found			
Main Memor	y Status		
Start	End	Occupied	PID
0		1	2
	9		
10	9 19		7
10 20		1	7 -1
	19	1	
20 	19	1 0 	

Removed Node! No Merging!

Main Memor	y Status		
Start	End	Occupied	PID
)	9	0	-1
10	19	1	7
20	19	0	-1
Current Pr	ocess at Qu	eue Head: 7	
Removed No	de! Merged	with Prev No	ode and Next Node!
Main Memor	y Status		
Start	End	Occupied	PID
O	19	0	-1

Best Fit:

Best Fit:			
Processe	s:		
PID	A.Time	B.Time	Memory Requirement
	5	5	8
l	4	7	8
2	9	3	5
3	1	10	1
4	1	7	5
5	4	8	7
6	9	4	4
7	0	1	2
8	0	3	2
9	1	2	5
Best Fit	Memory Allo	cation Algor	ithm
Current	Process at Q	ueue Head: 7	
Slot Fou	nd!		
Main Mem	ory Status		
Start	End	Occupied	PID
)	1	1	7
2	19	0	-1
Current	Process at Q	ueue Head: 8	
Slot Fou	nd!		
Main Mem	ory Status		
Start	End	Occupied	PID
)	1	1	7
2	3	1	8
1	19	0	-1

```
Current Process at Queue Head: 7
Removed Node! No Merging!
Main Memory Status
Start End
              Occupied PID
         19
Current Process at Queue Head: 3
Slot Found!
Main Memory Status
Start End Occupied PID
         19
Current Process at Queue Head: 4
Slot Found!
Main Memory Status
Start End Occupied PID
         19
Current Process at Queue Head: 9
Slot Found!
Main Memory Status
Start End
                  Occupied PID
         13
         19
Current Process at Queue Head: 8
Removed Node! Merged with Prev Node!
Main Memory Status
             Occupied PID
Start End
```

4	0	1	4
4	8	1	4
9	13	1	9
14 	19 	0 	-1
Current Pr		 t Queue Head: 9	
		ged with Next No	de I
Main Memory			αc.
Start		Occupied	PID
0	0	1	
1		0	
4	8		4
9	19	0	-1
Current Pro	ocess a	t Queue Head: 5	
Slot Found			
Main Memory	y Statu	S	
Start	End	Occupied	PID
0	0	1	3
1	3	0	-1
4	8	1	4
9	15	1	5
16	19	0	-1
Current Pro	ocess a	t Queue Head: 1	
Not Slot Fo	ound!		
Main Memory	y Statu	S	
Start	End	Occupied	PID
0	0	1	3
1	3	0	-1
4	8	1	4
9	15	1	5
16	19	0	-1
Current Pro	ocess a	t Queue Head: 0	
Not Slot Fo	ound!		

Main Memory Status

	_		
Start	End	Occupied	PID
0	0	1	3
1	3	0	-1
4	8	1	4
9	15	1	5
16	19	0	-1

Current Process at Queue Head: 4

```
Removed Node! Merged with Prev Node!
Main Memory Status
Start End
             Occupied PID
         15
16
         19
Current Process at Queue Head: 0
Slot Found!
Main Memory Status
Start End
                  Occupied PID
         15
16
         19
Current Process at Queue Head: 1
Not Slot Found!
Main Memory Status
Start End
                  Occupied PID
         15
16
         19
Current Process at Queue Head: 6
Slot Found!
Main Memory Status
              Occupied PID
Start
      End
         15
16
         19
20
         19
Current Process at Queue Head: 2
Not Slot Found!
Main Memory Status
Start End
                  Occupied PID
```

9 8 0 -1 9 15 1 5 16 19 1 6 20 19 0 -1	
16 19 1 6	
20 19 0 -1	
Current Process at Queue Head: 3	
Removed Node! No Merging!	
Main Memory Status	
Start End Occupied PID	
0 0 -1	
1 0	
9 8 0 -1	
9 15 1 5	
16 19 1 6	
20 19 0 -1	
Current Process at Queue Head: 2	
Not Slot Found!	
Main Memory Status	
Start End Occupied PID	
0 0 -1	
1 0	
9 8 0 -1	
9 15 1 5	
16 19 1 6	
20 19 0 -1	
Current Process at Queue Head: 1	
Not Slot Found!	
Main Memory Status	
Start End Occupied PID	
0 0 -1	
1 8 1 0	
9 8 0 -1	
9 15 1 5	
16 19 1 6	
20 19 0 -1	
Current Process at Queue Head: 5	
Removed Node! Merged with Prev Node!	
Main Memory Status	
Start End Occupied PID	
Bha Gocapica 115	

1	8	1	0
9	15	0	-1
16	19	1	6
20	19	0	-1
Current Pr	ocess at Qu	eue Head: 2	
Slot Found			
Main Memor	y Status		
Start	End	Occupied	PID
		0	
1	8	1	0
9	13	1	2
14	15	0	-1
16	19	1	6
20	19	0	
Current Pr	ocess at Qu	eue Head: 1	
Not Slot F			
Main Memor			
		Occupied	PID
	0		
1	8	1	0
9	13	1	2
14	15	0	-1
16	19	1	6
20	19	0	-1
Current Pr	ocess at Qu	eue Head: 6	
			de and Next Node!
Main Memor			
Start		Occupied	PID
0		0	
1	8		0
	13		
9		<u> </u>	
9 14	19	0	_1

Main Memory Status

Start	End	Occupied	PID
0	8	0	-1
9	13	1	2
14	19	0	-1

```
Current Process at Queue Head: 1
Slot Found!
Main Memory Status
Start End Occupied PID
        13
14
         19
Current Process at Queue Head: 2
Removed Node! Merged with Prev Node and Next Node!
Main Memory Status
Start End Occupied PID
        19
Current Process at Queue Head: 1
Removed Node! Merged with Next Node!
Main Memory Status
Start End Occupied PID
         19 0
```

Worst Fit:

Processes:			
PID	A.Time	B.Time	Memory Requirement
0	6	8	6
1	5	3	3
2	8	7	9
3	3	6	7
4	7	2	5
5	3	4	8
6	3	4	1
7	8	4	10
8	9	3	4
9	0	8	10
Best Fit Me	mory Alloca	tion Algorit	hm
Current Pro	cess at Que	ue Head: 9	
Slot Found!			
Main Memory	Status		
Start	End	Occupied	PID
0	9	1	9
10	19	0	-1

```
Current Process at Queue Head: 6
Slot Found!
Main Memory Status
Start
                    Occupied PID
10
          10
11
          19
Current Process at Queue Head: 3
Slot Found!
Main Memory Status
                    Occupied PID
Start
         End
10
          10
11
          17
18
          19
Current Process at Queue Head: 5
Not Slot Found!
Main Memory Status
Start End
                     Occupied PID
10
11
          17
18
          19
Current Process at Queue Head: 1
Not Slot Found!
Main Memory Status
Start
                     Occupied PID
10
11
          17
18
          19
Current Process at Queue Head: 0
Not Slot Found!
Main Memory Status
Start
                      Occupied
                                PID
0
10
          10
11
          17
```

18

19

```
Current Process at Queue Head: 6
Removed Node! No Merging!
Main Memory Status
Start End
                Occupied PID
10
         10
11
         17
18
         19
Current Process at Queue Head: 1
Not Slot Found!
Main Memory Status
Start
                    Occupied PID
10
11
         17
18
          19
Current Process at Queue Head: 4
Not Slot Found!
Main Memory Status
Start
                Occupied PID
10
11
18
          19
Current Process at Queue Head: 0
Not Slot Found!
Main Memory Status
Start
        End
                    Occupied PID
10
         10
11
          17
18
          19
Current Process at Queue Head: 5
Not Slot Found!
Main Memory Status
Start
                    Occupied PID
0
10
11
      17
```

18	19	0	-1
Current Pro	cess at Que	ue Head: 9	
Removed Nod	e! Merged w	ith Next Nod	e!
Main Memory	Status		
Start	End	Occupied	PID
0	10	0	-1
11	17	1	3
18	19	0	-1
Current Dre			
Current Pro Slot Found!	cess at Quei	ie neau; i	
Main Memory	Q+ 2+116		
		Occupied	חות
		1	
	10	0	
			3
	19		-1
		·	
Current Pro	cess at Que	ue Head: 4	
Slot Found!			
Main Memory	Status		
Start	End	Occupied	PID
0	2	1	1
3	7	1	4
8	10	0	-1
11	17	1	3
18	19	0	-1
Current Pro		ue Head: 0	
Not Slot Fo			
Main Memory			
		Occupied	
	2		
			4
	10		-1
	17		3
18	19 	0	-1
Current Pro	cess at Oue	ue Head: 5	
Not Slot Fo			
Main Memory			
_	End	Occupied	PID
0	2	1	1

3	7	1	4
8	10	0	-1
11	17	1	3
18	19	0	-1
Current Pro	ocess at Que	ue Head: 2	
Not Slot Fo			
Main Memory	y Status		
Start	End	Occupied	PID
		1	
3	7	1	4
8	10	0	-1
11	17	1	3
18	1.0	0	-1
Current Pro	ocess at Que	ue Head: 7	
Not Slot Fo			
Main Memory			
-		Occupied	PID
	2	1	
	_ 7	1	
8	10		-1
11	17	1	3
18	19	0	-1
			-
Current Pro	ocess at Que	ue Head: 3	
			de and Next Node!
Main Memory			
		Occupied	PID
		1	
	7		4
8	19		
	 		-
Current Pro	ocess at Que	ue Head: 8	
Slot Found			
Main Memory			
-		Occupied	PID
		1	
3	_ 7	1	4
0	11		<u> </u>

Current Process at Queue Head: 0

Slot Found!

12

```
Main Memory Status
Start
               Occupied PID
         End
         11
12
         17
18
         19
Current Process at Queue Head: 5
Not Slot Found!
Main Memory Status
Start End
                   Occupied PID
         11
12
         17
18
         19
Current Process at Queue Head: 2
Not Slot Found!
Main Memory Status
Start End
                   Occupied PID
         11
12
18
         19
Current Process at Queue Head: 7
Not Slot Found!
Main Memory Status
Start
               Occupied PID
         11
12
         17
18
         19
Current Process at Queue Head: 4
Removed Node! No Merging!
Main Memory Status
              Occupied PID
Start End
```

8 11 1 8 12 17 1 0 18 19 0 -1
12 17 1 0 18 19 0 -1
18 19 0 -1
Current Process at Queue Head: 5
Current Process at Queue Head: 5
Not Slot Found!
Main Memory Status
Start End Occupied PID
0 2 1 1
3 7 0 -1
8 11 1 8
12 17 1 0
18 19 0 -1
Current Process at Queue Head: 2
Not Slot Found!
Main Memory Status
Start End Occupied PID
0 2 1 1
3 7 0 -1
8 11 1 8
12 17 1 0
18 19 0 - 1
Current Process at Queue Head: 7
Not Slot Found!
Main Memory Status
Start End Occupied PID
0 2 1 1
3 7 0 -1
8 11 1 8
12 17 1 0
12 17 1 0 18 18 19 0 -1

Current Process at Queue Head: 1

Removed Node! Merged with Next Node!

Main Memory Status

Start	End	Occupied	PID
0	7	0	-1
8	11	1	8
12	17	1	0
18	19	0	-1

Current Process at Queue Head: 5

```
Slot Found!
Main Memory Status
Start End
              Occupied PID
         11
         17
12
18
         19
Current Process at Queue Head: 2
Not Slot Found!
Main Memory Status
Start
      End
               Occupied PID
         11
12
         17
18
         19
Current Process at Queue Head: 7
Not Slot Found!
Main Memory Status
Start End
               Occupied PID
8
         11
12
         17
18
         19
Current Process at Queue Head: 8
Removed Node! Merged with Prev Node!
Main Memory Status
Start End
             Occupied PID
         11
12
         17
18
         19
Current Process at Queue Head: 2
Not Slot Found!
Main Memory Status
               Occupied PID
Start End
```

11

		•	
12	17	1	0
18	19	0	-1
Current Pro	cess at Que	ue Head: 7	
Not Slot Fo	und!		
Main Memory	Status		
Start	End	Occupied	PID
	7		
	11	0	
		1	0
	19	0	-1
			<u> </u>
Current Pro	cess at Que	ue Head: 5	
Removed Nod	e! Merged w	ith Next Nod	le!
Main Memory	Status		
Start	End	Occupied	PID
	11		
	17	1	0
	19	0	
		· 	-
Current Pro	cess at Ollei	ie Head· 2	
Slot Found!	cess ac gae	ae neda. z	
	C+		
Main Memory		0	DID
	End	Occupied	LID
	8	1	
9	11		2 -1
9	11	0	-1
9 12 18 	11 17 19	0 1 0	-1 0
9	11 17 19	0 1 0	-1 0
9 12 18 	11 17 19 cess at Que	0 1 0	-1 0
9 12 18 Current Pro	11 17 19 cess at Quer und!	0 1 0	-1 0
9 12 18 Current Pro Not Slot Fo Main Memory	11 17 19 cess at Querund! Status	0 1 0	-1 0 -1
9 12 18 Current Pro Not Slot Fo Main Memory	11 17 19 cess at Querund! Status	0 1 0 ue Head: 7	-1 0 -1
9 12 18 Current Pro Not Slot Fo Main Memory Start 0	11 17 19 cess at Quetund! Status End	0 1 0 ue Head: 7 Occupied 1	-1 0 -1
9 12 18 Current Pro Not Slot Fo Main Memory Start 0 9	11 17 19 cess at Quer und! Status End 8	0 1 0 ue Head: 7 Occupied 1	-1 0 -1 PID 2
9 12 18 Current Pro Not Slot Fo Main Memory Start 0 9	11 17 19 cess at Quer und! Status End 8 11	0 1 0 1e Head: 7 Occupied 1 0	-1 0 -1 PID 2 -1
9 12 18 Current Pro Not Slot Fo Main Memory Start 0 9	11 17 19 cess at Quer und! Status End 8 11	0 1 0 ue Head: 7 Occupied 1 0 1	-1 0 -1 PID 2 -1 0
9 12 18 Current Pro Not Slot Fo Main Memory Start 0 9	11 17 19 cess at Quer und! Status End 8 11 17	0 1 0 ue Head: 7 Occupied 1 0 1	-1 0 -1 PID 2 -1 0
9 12 18 Current Pro Not Slot Fo Main Memory Start 0 9 12 18 Current Pro	11 17 19	0 1 0 Decupied 1 0 1 0 Head: 0	-1 0 -1 PID 2 -1 0
9 12 18 Current Pro Not Slot Fo Main Memory Start 0 9 12 18 Current Pro	11 17 19	0 1 0 Decupied 1 0 1 0 Head: 0	-1 0 -1 PID 2 -1 0 -1
9 12 18	11 17 19	0 1 0 ue Head: 7 Occupied 1 0 1 0 ue Head: 0 ith Prev Nod	-1 0 -1 PID 2 -1 0 -1
9 12 18	11 17 19	0 1 0 ue Head: 7 Occupied 1 0 1 0 ue Head: 0 ith Prev Nod	-1 0 -1 PID 2 -1 0 -1 -1

Current 1	Process at	Queue Head: 7	
Slot Four	nd!		
Main Memo	ory Status		
		Occupied	PID
		1	
		1	
19	19	0	-1
Current 1	Process at	Queue Head: 7	
Removed 1	Node! Merge	ed with Next No	de!
Main Memo	ory Status		
Start	End	Occupied	PID
		1	
		0	
	<u> </u>		
- -		^	
		Queue Head: 2	
		ed with Next No	ode!
Main Memo	ory Status		
Start	End	Occupied	PID
0	19	0	-1

- 2. Write a program that implements the following Page replacement algorithm.
- i) LRU (Least Recently Used)
- ii) Optimal Page Replacement algorithm

```
#include <bits/stdc++.h>
#define N 10
#define M 20
#define pi pair<int,int>
#define S second
#define F first
using namespace std;
void lru(vector<int>& required pages,int frame size) {
   cout<<"Least Recently Used Cache"<<endl;</pre>
  unordered map<int,int> umap;
   int page faults = 0;
   for(int k = 0;k<required pages.size();k++){</pre>
       int i = required pages[k];
       if(umap.count(i)){
           cout<<"Found Page: "<<i<<endl;</pre>
           umap[i] = T++;
       }else{
           page faults++;
           cout<<"Page not Found: "<<i<" (Page Fault) "<<endl;</pre>
           if(umap.size() == frame size) {
                int least recently used value = T, least recently used page = -1;
                for(auto i: umap) {
                    if(i.S<least recently used value){</pre>
                        least_recently_used_page = i.F;
                cout<<"Removed Page: "<<least recently used page<<endl;</pre>
                umap.erase(least recently used page);
           umap[i] = T++;
           cout<<"Inserted Page: "<<i<<endl;</pre>
       cout<<"Current Cache"<<endl;</pre>
       for(auto i: umap) {
           cout<<"("<<i.F<<","<<i.S<<")"<<" ";
```

```
cout << endl;
<<endl;
  cout<<"Total Page Faults: "<<page faults<<endl;</pre>
  cout<<endl<<endl;</pre>
void optimum(vector<int>& required_pages,int frame_size){
  #define MAX required pages.size();
  unordered map<int, int> umap;
  int page faults = 0;
  for(int k = 0;k<required pages.size();k++){</pre>
       int i = required pages[k];
       if(umap.count(i)){
           cout<<"Found Page: "<<i<<endl;</pre>
           umap[i] = MAX;
           for(int j =k+1;j<required pages.size();j++){</pre>
               if(required pages[j]==i){
                    umap[i] = j;
       }else{
           page faults++;
           cout<<"Page not Found: "<<i<" (Page Fault) "<<endl;</pre>
           if(umap.size() == frame size) {
               int optimum value = -1, least optimum page = -1;
               for(auto i: umap) {
                    if(i.S>optimum value){
                        optimum value = i.S;
                        least optimum page = i.F;
               cout<<"Removed Page: "<<least optimum page<<endl;</pre>
               umap.erase(least optimum page);
```

```
umap[i] = MAX;
           for(int j =k+1;j<required pages.size();j++){</pre>
                if(required pages[j]==i){
                    umap[i] = j;
           cout<<"Inserted Page: "<<i<<endl;</pre>
       cout<<"Current Cache"<<endl;</pre>
       for(auto i: umap) {
           cout<<"("<<i.F<<","<<i.S<<")"<<" ";
       cout<<endl;</pre>
<<endl;
   cout<<"Total Page Faults: "<<page faults<<endl;</pre>
int main(){
  srand(time(NULL));
  vector<int> required pages;
  for(int i = 0;i<M;i++){
       required_pages.push_back(rand()%N);
   lru(required_pages,3);
   optimum(required pages,3);
```

```
Page not Found: 3 (Page Fault)
Inserted Page: 3
Current Cache
Found Page: 0
Current Cache
(3,2) (5,1) (0,3)
Page not Found: 8 (Page Fault)
Removed Page: 5
Inserted Page: 8
Current Cache
(8,4) (3,2) (0,3)
Found Page: 3
Current Cache
(8,4) (3,5) (0,3)
-----
Page not Found: 6 (Page Fault)
Removed Page: 0
Inserted Page: 6
Current Cache
(6,6) (8,4) (3,5)
Page not Found: 2 (Page Fault)
Removed Page: 8
Inserted Page: 2
Current Cache
(2,7) (6,6) (3,5)
Found Page: 3
Current Cache
(2,7) (6,6) (3,8)
Page not Found: 7 (Page Fault)
Removed Page: 6
Inserted Page: 7
Current Cache
Page not Found: 5 (Page Fault)
Removed Page: 2
Inserted Page: 5
```

```
Current Cache
Page not Found: 8 (Page Fault)
Removed Page: 3
Inserted Page: 8
Current Cache
(8,11) (5,10) (7,9)
Page not Found: 0 (Page Fault)
Removed Page: 7
Inserted Page: 0
Current Cache
(0,12) (8,11) (5,10)
Page not Found: 9 (Page Fault)
Removed Page: 5
Inserted Page: 9
Current Cache
(9,13) (0,12) (8,11)
Page not Found: 1 (Page Fault)
Removed Page: 8
Inserted Page: 1
Current Cache
(1,14) (9,13) (0,12)
Page not Found: 7 (Page Fault)
Removed Page: 0
Inserted Page: 7
Current Cache
(7,15) (1,14) (9,13)
Page not Found: 8 (Page Fault)
Removed Page: 9
Inserted Page: 8
Current Cache
(8,16) (7,15) (1,14)
Found Page: 8
Current Cache
Page not Found: 9 (Page Fault)
```

```
Removed Page: 1
Inserted Page: 9
Current Cache
(9,18) (8,17) (7,15)
Page not Found: 1 (Page Fault)
Removed Page: 7
Inserted Page: 1
Current Cache
(1,19) (9,18) (8,17)
Total Page Faults: 16
Optimum Cache
Page not Found: 0 (Page Fault)
Inserted Page: 0
Current Cache
(0,3)
Page not Found: 5 (Page Fault)
Inserted Page: 5
Current Cache
(5,10) (0,3)
Page not Found: 3 (Page Fault)
Inserted Page: 3
Current Cache
(3,5) (5,10) (0,3)
Found Page: 0
Current Cache
(3,5) (5,10) (0,12)
Page not Found: 8 (Page Fault)
Removed Page: 0
Inserted Page: 8
Current Cache
(8,11) (3,5) (5,10)
Found Page: 3
Current Cache
(8,11) (3,8) (5,10)
Page not Found: 6 (Page Fault)
Removed Page: 8
```

```
Inserted Page: 6
Current Cache
(6,-1) (3,8) (5,10)
Page not Found: 2 (Page Fault)
Removed Page: 5
Inserted Page: 2
Current Cache
(2,-1) (6,-1) (3,8)
Found Page: 3
Current Cache
(2,-1) (6,-1) (3,-1)
Page not Found: 7 (Page Fault)
Removed Page: -1
Inserted Page: 7
Current Cache
(7,15) (2,-1) (6,-1) (3,-1)
Page not Found: 5 (Page Fault)
Inserted Page: 5
Current Cache
(5,-1) (7,15) (2,-1) (6,-1) (3,-1)
Page not Found: 8 (Page Fault)
Inserted Page: 8
Current Cache
Page not Found: 0 (Page Fault)
Inserted Page: 0
Current Cache
(0,-1) (8,16) (5,-1) (7,15) (2,-1) (6,-1) (3,-1)
Page not Found: 9 (Page Fault)
Inserted Page: 9
Current Cache
(9,18) (0,-1) (8,16) (5,-1) (7,15) (2,-1) (6,-1) (3,-1)
Page not Found: 1 (Page Fault)
Inserted Page: 1
Current Cache
```

```
Found Page: 7
Current Cache
(1,19) (9,18) (0,-1) (8,16) (5,-1) (7,-1) (2,-1) (6,-1) (3,-1)
Found Page: 8
Current Cache
(1,19) (9,18) (0,-1) (8,17) (5,-1) (7,-1) (2,-1) (6,-1) (3,-1)
Found Page: 8
Current Cache
(1,19) (9,18) (0,-1) (8,-1) (5,-1) (7,-1) (2,-1) (6,-1) (3,-1)
Found Page: 9
Current Cache
(1,19) (9,-1) (0,-1) (8,-1) (5,-1) (7,-1) (2,-1) (6,-1) (3,-1)
Found Page: 1
Current Cache
(1,-1) (9,-1) (0,-1) (8,-1) (5,-1) (7,-1) (2,-1) (6,-1) (3,-1)
Total Page Faults: 12
```