**LOVELY PROFESSIONAL UNIVERSITY**

**Academic Task-3 (Operating System)**

School of Computer Science and Engineering (Faculty of Technology & Sciences)

SUBMITTED BY

Name - Shekhar Anand

Reg no – 11803983

Roll no- B43

Section – K18KH

Email-id – [shekhar4321anand@gmail.com](mailto:shekhar4321anand@gmail.com)

Github link - <https://github.com/demin13/os_project>

SUBMITTED TO

Name of the faculty member: Keshav Dhir

Course code: CSE316

Course Title: Operating System

**DESCRIPTION**

Problem:

Ten students (a, b, c, d, e, f, g, h, i, j) are going to attend an event. There are lots of gift shops, they all are going to the gift shops and randomly picking the gifts. After picking the gifts they are randomly arriving in the billing counter. The accountant gives the preference to that student who has maximum number of gifts. Create a C or Java program to define order of billed students?

Solution:

This problem can be solved by using **priority scheduling** algorithm of operating system.

At billing counter all student will prioritized as per their number of gifts they picked and accountant will billed first who is having maximum number of gifts this can be done using priority algorithm of operating system and if same number of gifts will be there then in that case accountant will billed as per first come first serve basis(FCFS).

What is **priority scheduling?**

* Priority scheduling is a non-preemptive algorithm and one of the most common scheduling algorithms in batch systems.
* Each process is assigned a priority. Process with highest priority is to be executed first and so on.
* Processes with same priority are executed on first come first served basis.
* Priority can be decided based on memory requirements, time requirements or any other resource requirement.

**ALGORITHM AND COMPLEXITY**

Step 1: create 10 student arrays as n\_student, 20 shops array as n\_shops and 10 gifts array

as n\_gifts , create 10 billed for gifts as order\_billed.

Step 2: start loop from 0 🡨 i to 9 🡨 i

Step 3: Take random value for n\_shops as shop number from 0 to 50.

Step 4: Take random value for n\_gifts as number of gifts from 0 to 30.

**End of Loop of Step 2.**

Step 5: start loop from 0 🡨 k to 9 🡨 k

0 🡨 key, 0 🡨 index

Step 6: start loop from 0 🡨 l to 10-k 🡨 l

Step 7: if n\_gifts[l] >= key

Then: n\_gifts[l] 🡨 key

End of if structure.

Else: do nothing

**End of loop of Step 4.**

Step 8: key 🡨 n\_bill[k]

Step 9: search (n\_gifts, key, length(n\_gifts)) 🡨 index

Step 10: n\_student[index]🡨 order\_billed[k]

Step 11: delete (n\_gifts, index, length(n\_gifts))

Step 12: delete (n\_student, index, length(n\_student))

**End of loop of step 5**.

**Complexity**

Step 2: O(n)

Step 5: O(n)

Step 6: O(n)

**ADDITIONAL ALGORITHM AND COMPLEXITY**

**Search (array [], key, length)**

Step 1: Start loop from 0🡨 r to length🡨 r

Step 2: if key = array[r]

Then: r 🡨 index

Break

Else: do nothing

End of loop of Step 1.

Step 3: Return index.

Complexity: - Step 1: O(n)

**Delete (array [], index, length)**

Step 1: Start loop from 0 🡨 p to length 🡨 p

Step 2: if p = index

Step 3: Then: start loop from p 🡨 o to length 🡨 o

Step 4: array[o+1] 🡨 array[o]

End of loop of step 1.

Step 5: length-1 🡨 length;

Break

End of if structure.

Else: do nothing

End of loop of step 3.

Complexity: -

Step 1: O(n)

Step 3: O(n)

**Overall Complexity**: n + (n\*n) + n + (n\*n) = 2n +2(n^2) = **O(n^2)**

**CODE AND SNIPPET OF OUTPUT**

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#include<time.h>

#include<string.h>

//searching algorithm

int search(int array[],int keys,int leng){

int r,t;

for(r=0;r<leng;r++)

{

if(keys==array[r])

{

t=r;

break;

}

}

return t;

}

//deletion of any index from an integer array O(n^2)

void delete\_i(int arr[],int ind,int len)

{

int p,o;

for(p=0;p<len;p++)

{

if(p==ind)

{

for(o=p;o<len;o++)

{

arr[o]=arr[o+1];

}

len=len-1;

break;

}

}

}

//deletion of any index from an character array O(n^2)

void delete\_c(char arr[],int ind,int len)

{

int p,o;

for(p=0;p<len;p++)

{

if(p==ind)

{

for(o=p;o<len;o++)

{

arr[o]=arr[o+1];

}

len=len-1;

break;

}

}

}

int main(){

int i,j;

time\_t t;

srand(time(&t)); //uses of srand because every time random number will differ

char n\_student[]={'a','b','c','d','e','f','g','h','i','j'};

int n\_shops[20];

int n\_gifts[10];

for(i=0;i<10;i++) //O(n)

{

n\_shops[i]=rand()%50;

n\_gifts[i]=rand()%30;

}

printf("No. of gifts picked by which student and from which shop no :-\n\n");

for(j=0;j<10;j++)

{

printf("No of gifts picked by %c from shop no %d = %d\n",n\_student[j],n\_shops[j],n\_gifts[j]);

}

//At billing counter

int k,l,index=0;

int n\_bill[10];

int order\_billed[10];

for(k=0;k<10;k++)

{

int key=0;

for(l=0;l<10-k;l++)

{

if(n\_gifts[l]>=key)

{

key=n\_gifts[l];

}

}

n\_bill[k]=key;

index=search(n\_gifts,key,(sizeof(n\_gifts)/4));

order\_billed[k]=n\_student[index];

delete\_i(n\_gifts,index,(sizeof(n\_gifts)/4));

delete\_c(n\_student,index,(strlen(n\_student)));

}

printf("\nAt billing counter first preference will be given to those student those are\n");

printf("having maximum number of gifts and if equal then on FCFS basis:-\n\n");

for(i=0;i<10;i++)

{

printf("billing of student %c as he picked %d gifts.\n",order\_billed[i],n\_bill[i]);

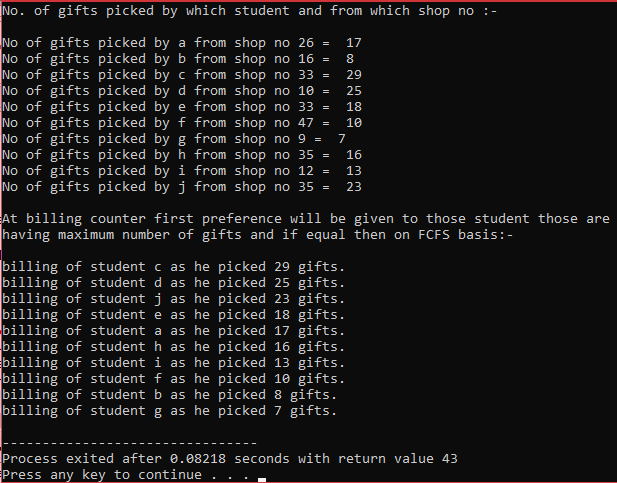
}

}

**SAMPLE** **TEST CASES AND ITS OUTPUT**

**Test case 1:-**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | **Expected output** | | | |
| **Student** | | **no\_of\_gifts** | **student\_order** | | **no\_gifts** | |
| **a** | | **17** | **c** | | **29** | |
| **b** | | **8** | **d** | | **25** | |
| **c** | | **29** | **j** | | **23** | |
| **d** | | **25** | **e** | | **18** | |
| **e** | | **18** | **a** | | **17** | |
| **f** | | **10** | **h** | | **16** | |
| **g** | | **7** | **i** | | **13** | |
| **h** | | **16** | **f** | | **10** | |
| **i** | | **13** | | **b** | | **8** |
| **j** | | **23** | | **g** | | **7** |

 **Actual output :-**

**Test Case 2:-**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | **Expected output** | | | |
| **Student** | | **no\_of\_gifts** | **student\_order** | | **no\_gifts** | |
| **a** | | **19** | **c** | | **29** | |
| **b** | | **21** | **g** | | **29** | |
| **c** | | **29** | **e** | | **25** | |
| **d** | | **5** | **f** | | **24** | |
| **e** | | **25** | **b** | | **21** | |
| **f** | | **24** | **a** | | **19** | |
| **g** | | **29** | **j** | | **17** | |
| **h** | | **13** | **h** | | **13** | |
| **i** | | **2** | | **d** | | **5** |
| **j** | | **17** | | **i** | | **2** |

**Actual output :-**

