NAME: Meet Raut

BATCH: S21

ROLL NO.: 2201084

Experiment 6 : Job Sequencing With Deadlines

AIM

To implement Job sequencing with deadlines using Greedy

THEORY

Job scheduling algorithm is applied to schedule the jobs on a single processor to maximise the profits.

Here-

- You are given a set of jobs.
- Each job has a defined deadline and some profit associated with it.
- The profit of a job is given only when that job is completed within its deadline.
- Only one processor is available for processing all the jobs.
- Processor takes one unit of time to complete a job.

The greedy approach of the job scheduling algorithm states that, “Given ‘n’ number of jobs with a starting time and ending time, they need to be scheduled in such a way that maximum profit is received within the maximum deadline”.

Approach to Solution

A feasible solution would be a subset of jobs where each job of the subset gets completed within its deadline.
Value of the feasible solution would be the sum of profit of all the jobs contained in the subset.
An optimal solution of the problem would be a feasible solution which gives the maximum profit.

Job Scheduling Algorithm

Set of jobs with deadlines and profits are taken as an input with the job scheduling algorithm and scheduled subset of jobs with maximum profit are obtained as the final output.

Algorithm

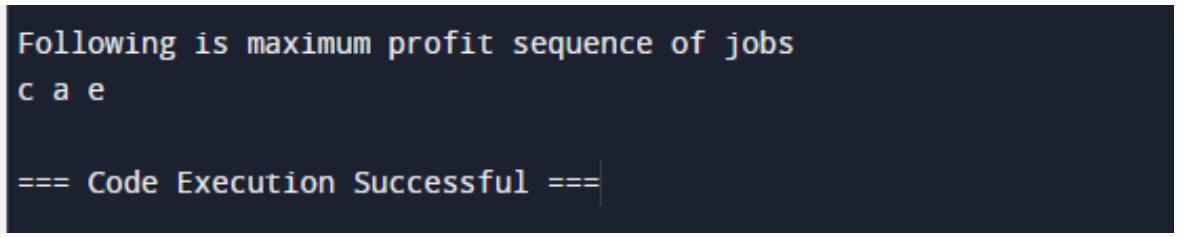
1. Step 1 − Find the maximum deadline value from the input set of jobs.
2. Step 2 − Once, the deadline is decided, arrange the jobs in descending order of their profits.
3. Step 3 − Selects the jobs with highest profits, their time periods not exceeding the maximum deadline.
4. Step 4 − The selected set of jobs are the output.

CODE

```
#include <stdbool.h>
#include <stdio.h>
#include <stdlib.h>

typedef struct Job
{

char id; int dead; int profit;

} Job;

int compare(const void *a, const void *b)
{
Job *temp1 = (Job *)a; Job *temp2 = (Job *)b;
return (temp2->profit - temp1->profit);
}

int min(int num1, int num2)
{
return (num1 > num2) ? num2 : num1;
}

void printJobScheduling(Job arr[], int n)
{
qsort(arr, n, sizeof(Job), compare); int result[n];
bool slot[n];

for (int i = 0; i < n; i++) slot[i] = false;

for (int i = 0; i < n; i++)
{

for (int j = min(n, arr[i].dead) - 1; j >= 0; j--)
{

if (slot[j] == false)
{
result[j] = i; slot[j] = true; break;
}
}
}
```

```
for (int i = 0; i < n; i++) if (slot[i])
printf("%c ", arr[result[i]].id);
}

int main()
{Job arr[] = {{'a', 2, 100},
{'b', 1, 19},
{'c', 2, 27},
{'d', 1, 25},
{'e', 3, 15}};
int n = sizeof(arr) / sizeof(arr[0]); printf(
"Following is maximum profit sequence of jobs \n");

printJobScheduling(arr, n); return 0;
}
```

OUTPUT



Conclusion

Hence we have successfully studied and implemented job sequencing with deadlines using greedy algorithm.