Week 4

E04-01. Implement algorithms of interval scheduling and give some examples to test it.

Input: n jobs, start time and finish time of each job

Output: maximum subset of mutually compatible jobs.

Example:

Input:

5
0 2
1 4
3 5
4 7
5 6

Output:

3

E04-02. Implement algorithms of interval partitioning and give some examples to test it.

Input: n lectures, start time and finish time of each lecture

Output: the minimum number of classrooms to schedule all lectures.

Example:

Input:

5
0 2
1 4
3 5
4 7
5 6

Output:

2

E04-03. Implement the algorithms of scheduling to minimize lateness and give some examples to test it.

Input: n jobs, processing time and deadline time of each job

Output: job scheduling to minimize maximum lateness.

Example:

Input:

5
1 9
3 6
4 9
2 8
3 14

Output:

1

Week 5

Exercises

E05-01. Implement algorithms of optimal caching and give some examples to test it.

Input:

The first line is size of cache k, initial blocks number n in cache and the number s for the sequence of requests.

The second line is the initial block no in cache

The third line is the block no for the sequence of request.

Output: eviction schedule(when some blocks can be considered at the same time, the longest not used one should be evicted first)

Example:

Input:

3 1 10

1

2 4 3 1 5 3 2 1 4 2

Output:

4 1 5 3

1

1 2 4

1 2 3 删除4 1 2 3

1 2 3

5 2 3 删除1 5 2 3

1 2 3 删除5 删除最大的？

1 2 4 删除3

3 2 1 5

3 4

1 2 3 4 1 2 5 3 6 2 3 1 7 4 1

1 4 1 5 6 3 7

3 4 1

3 4 2

3 1 2

3 5 2

3 6 2

3 2 1

7 2 1

4 2 1

E05-02. Implement Dijkstra algorithms of single-source shortest path and give some examples to test it.

Input: a directed graph with n nodes and e edges, source node s, the length of each edge (x_i, x_j, l)

Output: the shortest distance of other nodes and corresponding path.

Example:

Input:

5 8 1

1 2 2

2 3 2

2 4 1

1 3 5

3 4 3

1 4 4

1 5 7

4 5 2

Output:

0 2 4 3 5

8 16 1

1 5 9

1 2 5

1 8 8

5 8 5

5 6 4

2 8 4

8 6 6

2 3 12

2 4 15

8 3 7

6 3 1

3 4 3

3 7 11

5 7 20

6 7 13

4 7 9

5 8 1

1 2 2

2 3 2

2 4 1

1 3 5

3 4 3

1 4 4

1 5 7

4 5 2

7 12 1

2 0 4

2 5 5

0 1 2

0 3 1

1 4 10

1 3 3

3 2 2

3 5 8

3 6 4

3 4 2

4 6 6

6 5 1

E05-03. Implement algorithms of minimum spanning tree and give some examples to test it.

Input: a undirected graph with n nodes and e edges, the length of each edge(x_i, x_j, l)

Output: the sum of all edges in minimum spanning tree.

Example:

Input:

```
5 8
1 2 2
2 3 2
2 4 1
1 3 5
3 4 3
1 4 4
1 5 7
4 5 2
```

Output:

```
7
```

Week 6

Exercises

E06-01. Implement Kruskal's algorithm of minimum spanning tree and give some examples to test it. (If you implement Kruskal in homework5, please implement Prim)

Input: a undirected graph with n nodes and e edges, the length of each edge(x_i, x_j, l)

Output: the sum of all edges in minimum spanning tree.

Example:

Input:

```
5 8
1 2 2
2 3 2
2 4 1
1 3 5
3 4 3
1 4 4
1 5 7
4 5 2
```

Output:

```
7
```

E06-02. Implement greedy algorithms of Huffman codes and give some examples to test it.

Input: The first line is the number n for symbols. The second line is the frequency of the symbols.

Output: The avera e len th of Huffman codes.

Example:

Input:

```
4
1 2 3 4
```

Output:

```
1.90
```