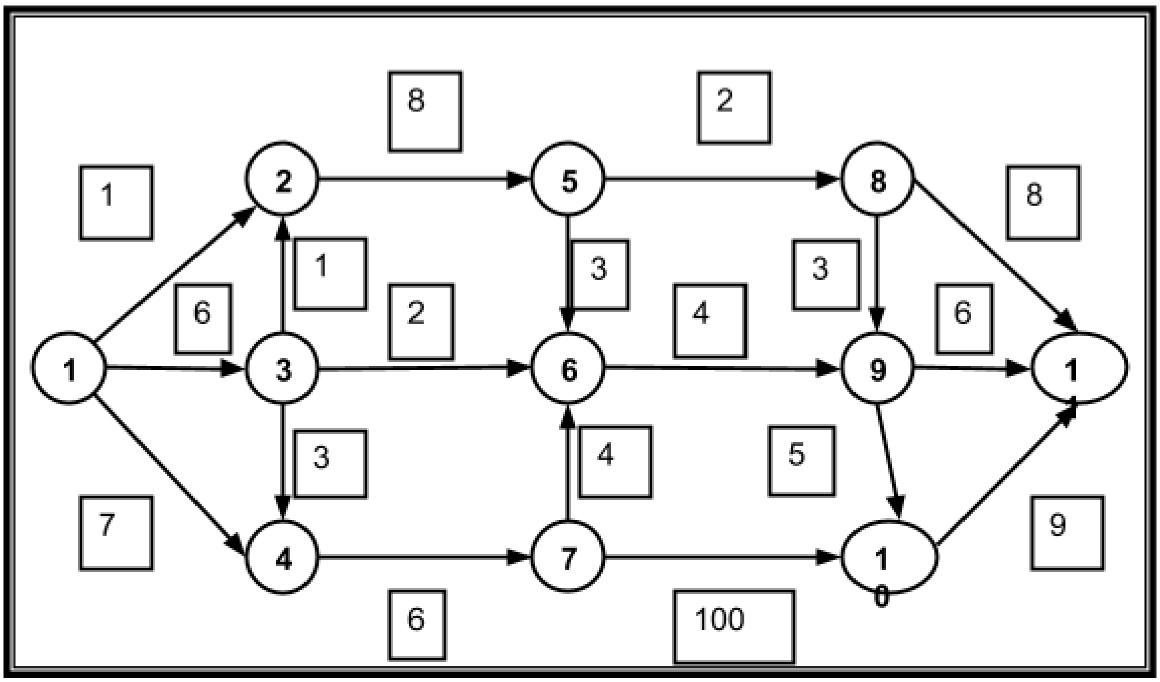
EC440: HW6

1. Consider the graph in Figure 1 as a directed, capacitated graph, where the numbers indicate an arc's capacity to carry from node 1 to node 11. In the max-flow algorithm of Ford and Fulkerson, the key step is, once a path has been found, to augment the flow and construct the residual graph for the next iteration.

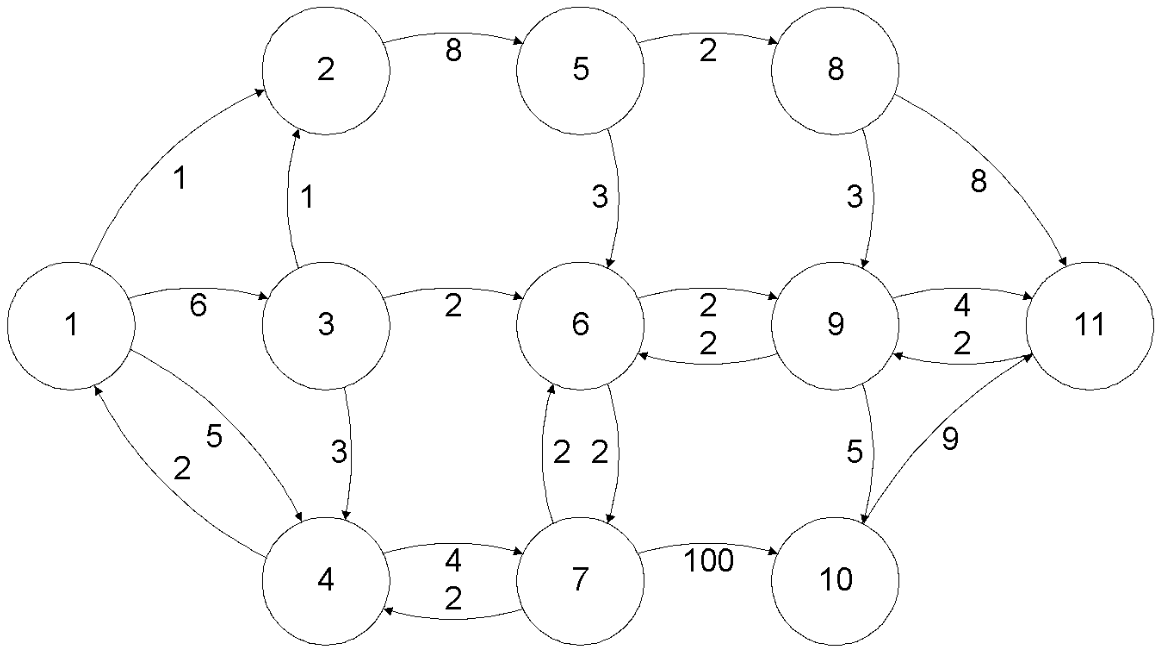


(a) Consider the path 1 - 3 - 4 - 7 - 10 - 11. What is the capacity of this path?

Solution: 3

(b) Suppose we send two units of flow along path 1 - 4 - 7 -6- 9-11. Construct the residual graph which remains after this has been sent.

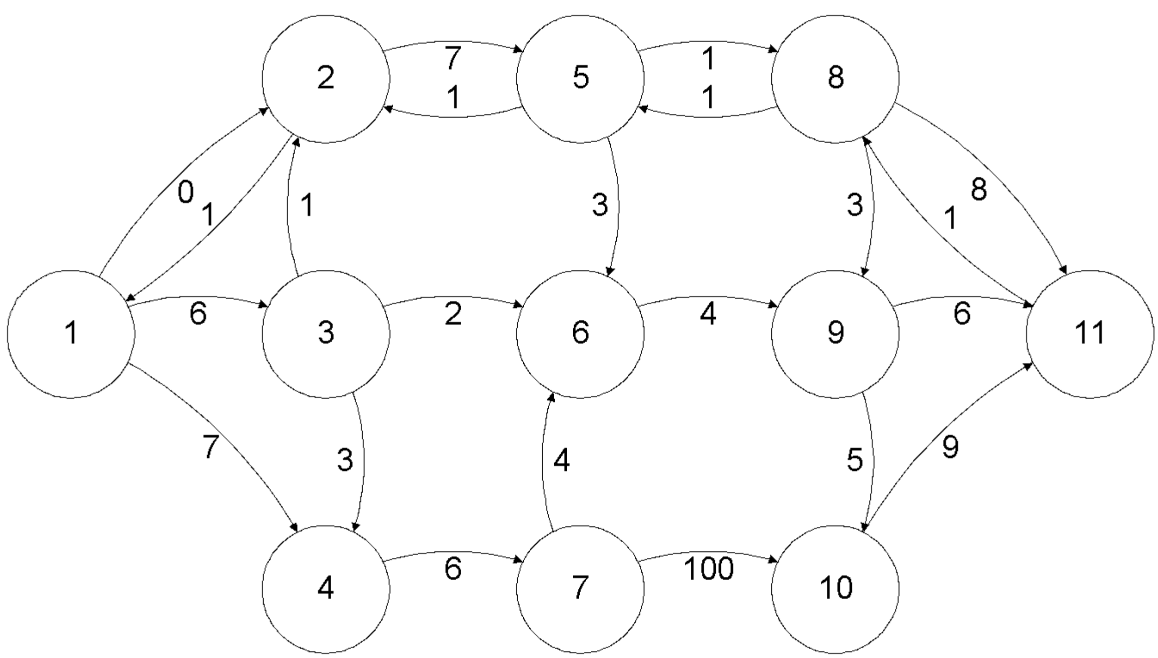
Solution:



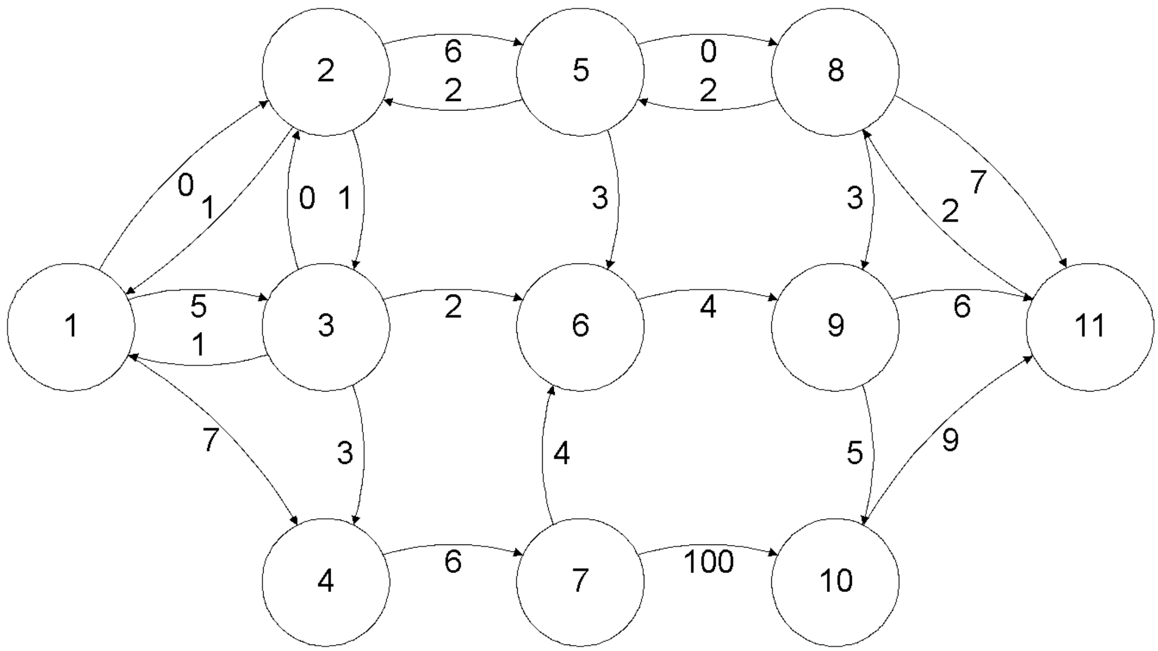
(c) Find the maximum flow from node 1 to node 11 in this graph and compare it with the min cut. (SHOW the residual graph after each augmentation. The last page has some blank figures to use for the residual and flow diagrams.)

<u>Solution:</u>

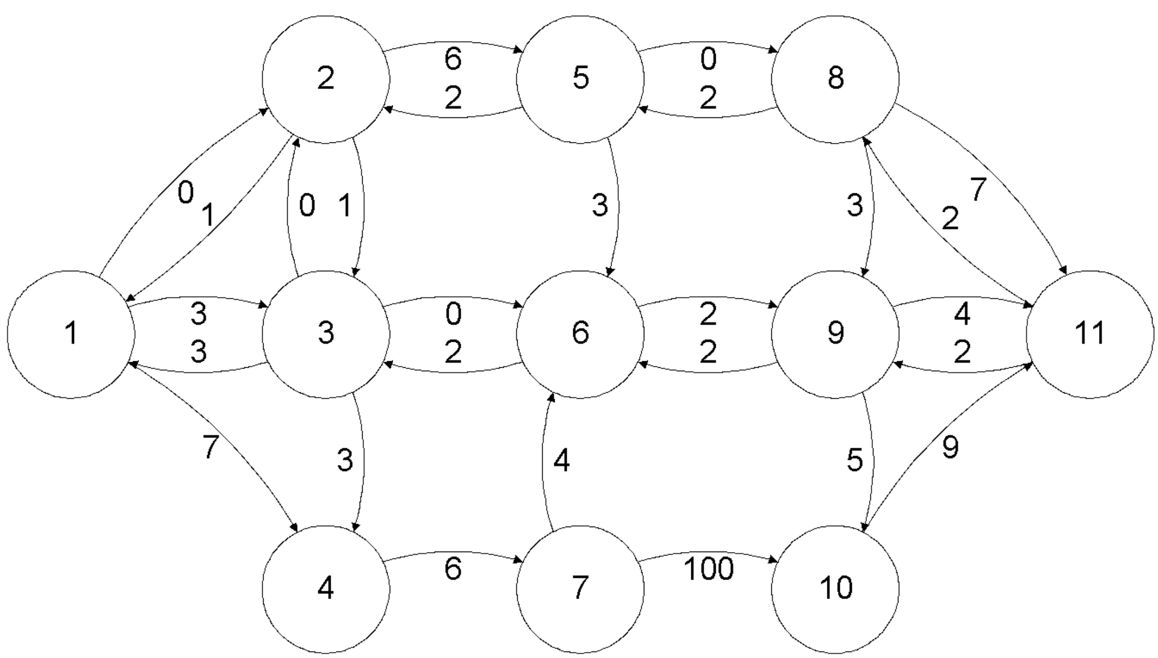
Path 1-2-5-8-11 Flow 1



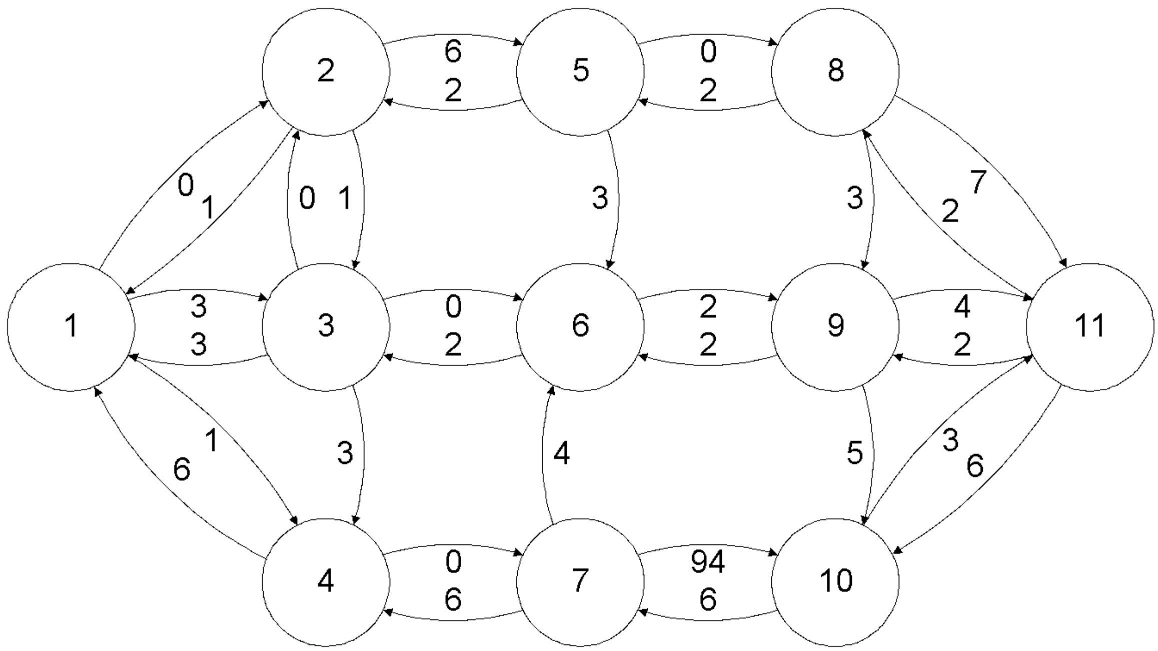
Path 1-3-2-5-8-11 Flow 1



Path 1-3-6-9-11 Flow 2



Path 1-4-7-10-11 Flow 6



Max flow thus is 1+1+2+6=10 through the cut {1,2},{2,3},{3,6},{4,7}

2. Assume you have a scheduling problem with 10 customers. Customer i has value Vi, and requires processing time Ti.

The values of Vi and Ti are listed below as arrays:

Assume that jobs can be scheduled partially, so that a job of value Vi which requires time Ti will receive value t*(Vi/Ti) if processed only for time t.

(a) Find the maximum value which can be scheduled with total processing time 30 units.

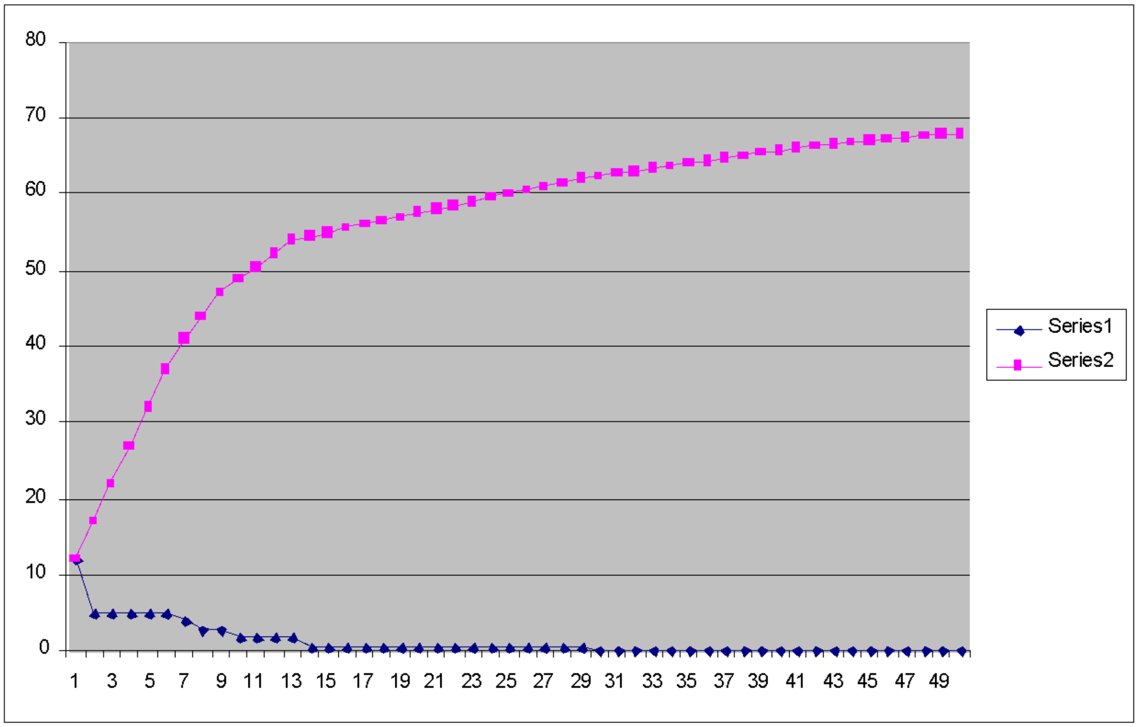
Solution: assign weights (values per time unit) W and indices I to the jobs.

V = {	6,	25,	4,	1,	7,	3,	12,	1,	1,	8 }
T = {	2,	5,	1,	4,	4,	9,	1,	5,	3,	16 }
W=	3	5	4	¼	7/4	1/3	12	1/5	1/3	½
I=	1	2	3	4	5	6	7	8	9	10

Then place the jobs in the decreasing value order:

I(index)	7	2	3	1	5	10	6
T(time)	1	5	1	2	4	16	1
V(value)	12	25	4	6	7	8	1/3
AT(accumulated time)	1	6	7	9	13	29	30
AV(accumulated value)	12	37	41	47	54	62	62+1/3

(b) Let T denote the total processing time, and let V (T) denote the maximum value which can be scheduled in processing time T . Plot V (T) for T = 1,…,50 (Hint: Use a spreadsheet or a program.)



3. Assume you have a scheduling problem with 10 customers. Customer “i” has value Vi, and must be processed before deadline Ti.

The processing time of each customer is exactly one unit of time. The values of Vi and Ti are listed below as arrays:

V = {	2,	4,	3,	1,	3,	5,	7,	3,	4,	6}
T = {	9,	6,	4,	3,	7,	10,	1,	3,	2,	2}
I	1	2	3	4	5	6	7	8	9	10

Solution:
The idea is the same as in #2 but after having been sorted the jobs are scheduled for their latest possible execution time.

V	7	6	5	4	4	3	3	3	2	1
T	1	2	10	6	2	7	4	3	9	3
I	7	10	6	2	9	5	3	8	1	4

Find the optimal sequence of jobs which can be scheduled in order to complete as much value as possible.

Time	0	1	2	3	4	5	6	7	8	9
I	7	10	8	3		2	5		1	6

Total Value = 33

4. Assume you have a scheduling problem with 10 customers. Customer "i" has processing time Ti. The value of each customer is the same. The objective is to minimize the sum across all customers of the completion time of each customer. The values of Ti are listed below as an array:

T = {	3,	4,	5,	10,	2,	6,	1,	1,	8,	2}
I	1	2	3	4	5	6	7	8	9	10

Find the optimal sequence of jobs to minimize the sum of the completion times.

Solution:

The jobs need to be scheduled in the order of increasing running time

7 8 5 10 1 2 3 6 9 4

The sum of the completion times is

1 + 2 + 4 + 6 + 9 + 13 + 18 + 24 + 32 + 42 = 151

5. Suppose that, instead of final exams, you were assigned a "project Friday", where each of your courses gives you a different project at 8 AM. Unfortunately, you are trying to get 2 degrees in a total of 3 years, so you have a slight overload of 7 courses this semester (!!!) Each project requires exactly 2 hours of work, and has a due time which is different. The instructors will give you no credit if it is late by 1 second, or if you work less than the required 2 hours. According to your subjective value rating, the value of turning each project in on time is given below, along with the deadline for the project. :

Project:	1	2	3	4	5	6	7
Value:	6	3	5	3	4	2	7
Deadline:	2pm	12noon	12noon	4pm	2pm	4pm	6pm

Find an optimal schedule for turning in your projects, starting work at 8 AM on Friday, and determine the value achieved by that schedule.

Solution:

The logic is the same as in #3

Start time:	8am	10am	12noon	2pm	4pm
Project:	5	3	1	2 or 4	7

The projects 6 and 4 or 2 cannot be completed. Total value is 25.

6. Consider the following scheduling problem: There are 10 tasks, each of which require a certain amount of processing time and have a value, as illustrated in the table below:

Task:	1	2	3	4	5	6	7	8	9	10
Value:	2	3	4	5	6	7	6	5	4	3
Processing Time:	1	2	3	4	5	6	7	6	5	4

(a) Suppose that there is a single machine with total capacity of 23 units of time, and that one gets partial credit for partially processing a task, so that processing a task of value Vi for time t when the requirements are Ti units provides value t(Vi/Ti). What is the optimal set of tasks to schedule, and the value achieved by this optimal schedule?

Solution: again the value per time unit is being maximized

Task:	1	2	3	4	5	6	7	8	9	10
Value:	2	3	4	5	6	7	6	5	4	3
Value per time unit:	2	3/2	4/3	5/4	6/5	7/6	6/7	5/6	4/5	¾
Processing Time:	1	2	3	4	5	6	7	6	5	4

And the jobs are scheduled with decreasing value as follows

Task:	1	2	3	4	5	6
Value:	2	3	4	5	6	3.5
Value per time unit:	2	3/2	4/3	5/4	6/5	7/6
Processing Time:	1	2	3	4	5	6

The value is 2+3+4+5+6+7 + (13/7) = 28 5/7

(b) Suppose that there are 3 machines: Machines 1 has capacity 10, Machine 2 has capacity 11, and Machine 3 has capacity 13. Suppose also that there is no partial credit, so that you only receive value for a task if it is processed fully. Find an optimal set of tasks to schedule on each machine, and the total value scheduled across all of the machines. (Hint: Think really hard before you resort to computation...)

Solution:

Capacity:	Job/Time/Value			Value
10	2/2/3	3/3/4	5/5/6	13
11	1/1/2	4/4/5	6/6/7	14

13	7/7/6	6/6/5		11

Got lucky total value is 38.

7. Consider the problem of finding the string "He is a scary @#$!&$#ing moron" in next Sunday's New York Times, which is available to you in electronic form. The Times is available uncompressed, with each letter represented by a byte, using the standard ASCII character codes for numbers 0 to 127. You decide to use the KMP search algorithm to find all occurrences of "moron" in binary format.

Compute the prefix function for the binary expansion of the word moron. The ASCII character codes for capital letters A-Z start at 65 increasing sequentially to 90 and for lower case letters a-z start at 97 increasing sequentially to 122. Thus, the ASCII codes for the letters are: m = 109, o =111, r = 114, n = 110 in decimal or m = 155, o = 157, r = 162, n = 156 in octal. (If you don't believe me or just for fun run):

```
#include <stdio.h>
#define CHAR_SET 256
int main(void)
{
        int i;
        for(i=0; i<CHAR_SET; i++)
                printf(``%d-th ASIC character, %c, is %o in octal\n",i,i,i);
        return 0;
}
```

0110110101101111011100100110111101101110

q	1	2	3	4	5	6	7	8	9	10
Pi(q)	0	0	0	1	2	3	4	5	1	2
q	11	12	13	14	15	16	17	18	19	20
Pi(q)	3	4	5	6	0	0	1	2	3	0
q	21	22	23	24	25	26	27	28	29	30
Pi(q)	1	1	2	1	1	2	3	4	5	6
q	31	32	33	34	35	36	37	38	39	40
Pi(q)	0	0	1	2	3	4	5	6	0	1

8. Consider finding the median of N elements in an unsorted list.

(a) How may partitions does it take in worst case to do quick sort and select the N/2 th element? What is the average O(f(N)) of this algorithm.

Solution:
Worst case N^2

Average and best case NlogN

(b) The O(N) worst case Median ending algorithm describe in class (and the online web notes) use a recursive algorithm to find an approximate median. Below is the list of 75 numbers divided into 15 lists of 5. Find the approximate median by recursion from the list down to a 15 elements divided into 3 lists of 5. Show the intermediate tables clearly.

Solution:

9	12	83	23	40	61	27	22	13	60	8	81	5	1	19
11	40	99	23	47	71	20	8	78	71	18	16	22	81	67
81	83	24	9	21	3	40	30	48	8	85	8	58	46	85
10	26	71	17	2	34	71	23	91	40	27	24	34	2	21
13	24	41	7	32	4	18	24	19	19	10	2	87	3	65

Sort each column

9	12	24	7	2	3	18	8	13	8	8	2	5	1	19
10	24	41	9	21	4	20	22	19	19	10	8	22	2	21
11	26	71	17	32	34	27	23	48	40	18	16	34	3	65
13	40	83	23	40	61	40	24	78	60	27	24	58	46	67
81	83	99	23	47	71	71	30	91	71	85	81	87	81	85

Divide middle row into 3 fives and sort them.

11	26	71	17	32
11	17	26	32	71

34	27	23	48	40
23	27	34	40	48

18	16	34	3	65
3	16	18	34	65

Finally sort 26, 34 and 18 to get the approximate median of 26.