# Exam INF102@UiB, fall 2012
# - An unofficial translation

Markus Dregi

No auxiliary resources allowed. You do not have to give java code in your solutions, pseudo code or a description is sufficient. Justify all your answers!

## Task 1

Give the running time of each of the code snippets below as order of growth. You can assume $n$ to be a positive integer. In c) and d) you are to give the running time of the function F.

a)
```
for(i = 1;i < n;i = i * 2)
  for(j = 0;j < n;j++)
    for(k = 0;k < j;k++)
      sum += A[j][k] * i;
```

b)
```
i = 0;
while (i < n)
    if (i % 2 == 0)
        i = i + 3;
    else
        i = i  1;
```

c)
```
F(i) {
    if (i == 1)
        return i;
    return F(i/2) + F(i/2);
}
```

d)
```
F(i) {
    if (i == 1)
        return i;
    return F(i-2) + F(i-2);
}
```

## Task 2

a) Demonstrate bottom-up merge sort on the following sequence of integers:

34, 12, 89, 8, 4, 45, 41, 9, 32, 75, 10, 34, 3, 21, 55, 43.

b) Derive the running time of merge sort.

c) For each of the cases below you are to find an appropriate sorting algorithm:

1: Each element is at most 10 positions away from its final position.

2: Each element is a number between 1 and 1000.

3: You want the sorting algorithm to be fast in most cases.

4: You want a sorting algorithm that is fast in the worst case and that also does not need to store more than a couple of extra variables in memory.

5: You want a sorting algorithm that is as fast as possible, but also stable.

6: You want a sorting algorithm that moves the elements as little as possible.

## Task 3

We are to study symbol tables in this task. Note that you are not supposed to use hashing!

a) Describe and analyze data structures and algorithms were the slowest of the two operations `put(Key key, Value val)` and `contains(Key key)` is as fast as possible.

b) The union of two symbol tapes $S_1$ and $S_2$ contains all the elements that are either in $S_1$ or $S_2$, while the intersection of $S_1$ and $S_2$ contains all the elements that are in both. For an example, lets say that $S_1$ contains the keys $\{A, B, D\}$ and $S_2$ the keys $\{C, B, D\}$. Then the union of $S_1$ and $S_2$ contains the keys $\{A, B, C, D\}$ and the intersection $\{C, D\}$.

Assume that $S_1$ and $S_2$ are represented by the data structures given in a). Describe and give the running time for two methods that implements the two operations `union` and `intersection`.

c) Ignore the work you have done so far in the task. Describe and analyze data structures and algorithms that implements the operations `union` and `intersection`, such that their running time is as fast as possible.

d) Describe and analyze algorithms that implements the operations `put` and `contains` for the data structures you gave in c).

## Task 4

a) Describe and analyze an effective algorithm that decides whether a directed graph is acyclic.

b) Demonstrate the algorithm on the graph in the figure. Whenever you have multiple choices, process the vertices in lexicographic order. For the figure, we refer to the Norwegian version.

c) Use the algorithm from a) to give a topological sorting of the vertices in the graph. Once again, we break ties by taking the lexicographically first vertex.

d) Describe and analyze an efficient algorithm that decides if a directed acyclic graph has a path that visits all of the vertices exactly once.