Computer Science 3711 (Fall 2005):
Assignment #1
Due: 4:00 PM on Wednesday, September 24, 2005

1. **(15 marks)** For each of the situations described below, state the corresponding computational problem in terms of its inputs and outputs along the lines of the statement of the sorting problem on page 5 of the textbook. Where possible, please phrase the entities in these problems in terms of abstract mathematical structures, *e.g.*, sets, functions, graphs, nodes, edges.

   (a) **(5 marks)** An air freight shipping company, Xanadu Airways, is offering worldwide freight container shipping. There is a standard set S of different types of shipping containers, and a shipping order is a subset of S. Each airplane in Xanadu Airway's fleet has an associated cargo specification that is a subset of S; note that a plane need not carry all of its cargo specification each time it makes a delivery. A shipping order can be filled if there is a set of airplanes that, between them, can carry all of the containers specified in that order. Given a particular shipping order and an integer $k > 0$, the company would like to know whether or not that order can be filled using at most k airplanes.

   (b) **(5 marks)** An air freight shipping company, Yolanda Airways, is offering worldwide flower-seed shipping. Each shipping order consists of a set of bins of different kinds of flower seeds, and each bin of seeds has a specified weight in kilograms. Each airplane in Yolanda Airway's fleet can carry up to $l > 0$ kilograms of seeds. A shipping order can be filled if there is a set of airplanes that, between them, can carry all bins of seeds in the order such that no bin of seeds is split over two or more airplanes. Given a particular shipping order, the company would like to know the smallest number of airplanes that can be used to fill that order.

   (c) **(5 marks)** An air freight shipping company, Zulu Airways, is offering worldwide toxic-waste shipping. The company is trying to hire pilots for its airplanes, and in their interviews, each pilot has specified a set of requirements, any one of which will make them accept the job, *e.g.*, monthly Hawaiian vacations, company sports cars, unlimited liquor expense accounts. The Human Resources (HR) Department at Zulu Airways is chronically understaffed, and hence will only handle the smallest number of different types of pilot job-requirements possible, regardless of how many pilots are hired. Given a particular set S of pilots and their job requirements, the company would therefore like to know a set of job-requirements that will simultaneously ensure that (1) all pilots in S will accept working for Zulu Airways and (2) the HR department will handle the associated job-requirement administration workload.

2. **(20 marks)** For each of the algorithms below, derive an **exact** time complexity function $T(n)$.

   (a) **(10 marks)**

```
sum = 1
for i = 1 to n do
    sum = sum + i
    for j = 1 to n * n do
        k = i * j
        k = k * k * k * k
        sum = sum / (k + 13)
    sum = sum / i
sum = sum + 25
```

   (b) **(10 marks)**

```
sum = 1
for i = 1 to n do
    sum = sum / i
    k = i * i
    for j = 1 to n do
        sum = sum * (k - 13)
    for j = 1 to n do
        sum = sum * (k - 13)
sum = sum * 14 * 10000000
for i = 1 to n * n do
    sum = sum + i
sum = sum - 42
```

3. **(20 marks)** For each of the algorithms below, derive a **worst-case** time complexity function $T(n)$.

   (a) **(10 marks)**

```
sum = 13
cond = false
for i = 1 to n do
    for j = 1 to 7 do
        sum = sum / (i + j)
    if COND(sum)
        cond = true
    else
        for j = 1 to n do
            sum = sum - j
if cond
    for j = 1 to n * log(n) do
        sum = sum + (i/j)
```

Note that method `COND()` runs in 5 timesteps and method `log(n)` returns the logarithm (base 2) of n, *i.e.*, $\log_2 n$. .

(b) **(10 marks)**

```
i = 1
sum = 57
finished = false
while ((i <= n) and (not finished))
    for j = 1 to i do
        if (COND(i))
            sum = sum + (i/j)
            for k = 1 to log(n) do
                sum = sum + k
        else
            sum = sum - (j/i)
    if COND(sum) then
        finished = true
        i = 57
    else
        i = i + 1
sum = sum / i + 63
```

Note that method `COND()` runs in 4 timesteps.

4. **(10 marks)** Give the asymptotic worst-case, *i.e.*, Big-Oh, time complexity functions $O(f(n))$ for each of the algorithms given in Questions #2 and #3 above.

5. **(20 marks)** For each of the algorithms below, derive an **asymptotic worst-case**, *i.e.*, Big-Oh, complexity function $O(f(n))$. Briefly explain the reasoning behind each derivation.

(a) **(5 marks)**

```
sum = 157
cond = false
for i = 1 to 7 do
    for j = 1 to n do
        sum = sum * (i/j)
        if (sum > 23)
            sum = sum + 23
            k = sum
            sum = sum + k
        else
            k = sum - 23
            for k = 1 to log(n) * n do
                sum = sum - (k/j)
if (cond)
    sum = sum - 256
```

(b) **(5 marks)**

```
sum = 42
for i = 1 to n * n do
    j = 1
    finished = true
    while ((j <= n) and (not finished)) do
        for k = 1 to log(n) * n do
            sum = sum / (k * i) + j
        if COND(sum)
            finished = true
```

Note that method `COND()` runs in $(n + 13)$ timesteps.

(c) **(5 marks)**

```
sum = 42
for i = 1 to n * n do
    j = 1
    finished = false
    while ((i <= n) and (not finished)) do
        for k = 1 to log(n) * n do
            finished = true
        if COND(sum)
            sum = sum / (k * i) + j
```

Note that method `COND()` runs in $(n + 13)$ timesteps.

(d) **(5 marks)**

```
sum = 42
for i = 1 to n * log(n) do
    j = 1
    finished = false
    for k = 1 to  n do
        if COND(sum)
            sum = sum / (k * i) + j
        while ((j <= n) and (not finished)) do
            finished = true
```

Note that method `COND()` runs in $(n + 13)$ timesteps.

6. **(15 marks)** Prove or disprove the following:

   (a) **(5 marks)** $f(n) = (n - 2)(n - 6)$ is not $\Theta(n^2)$.

   (b) **(5 marks)** $f(n) = n^d + 10n^2$, where d is some integer constant greater than or equal to 2, is $O(n^d)$.

   (c) **(5 marks)** $f(n) = 10^{127}2^n$ is $\Omega(3^n)$.