1. (10%) True or False (you must defend your answer): the following nondeterministic algorithm solves the No Partion Problem in nondeterministic polynomial time.

```
s:=0
for i:=1 to n do
   j:=choice({0,1})
   if j=1 then
      s:=s ∪ {i}
   endif
endif
if ∑_{i∈s} c_i ≠ ∑_{i∉s} c_i
then
   success, print("yes")
else
   failure, print("no")
endif
```

2. (20%) Trees such as binary tree, B-tree, etc. are normally used to store a set of values in a hierarchical manner so as to speedup search process. However, in some circumstances each of the values in a set may be a value range represented by a pair of two values. For example,

   {(2, 5), (3, 4), (1, 6), (4, 7)}.

   Can you suggest a way to store pairs of values so as to speedup search process in this situation? Describe your method in detail.

   Hints:

   - Search conditions in this situation can be like
     "Find the pairs that contain the point value 4.5"
     "Find the pairs that overlap with the range (3.4, 6.1)", etc.
   - You may start from trying to modify a binary tree for this purpose.

3. (18%) Let $A = (a_1, a_2, ..., a_n)$ and $B = (b_1, b_2, ..., b_m)$ be two sets. Assume $1 \leq a_i \leq p, 1 \leq i \leq n$ and $1 \leq b_i \leq p, 1 \leq i \leq m$. All a_i's and b_i's are integers. Write an algorithm to determine whether A and B are equal. Your algorithm should work in O(n+m) time.

4. (20%) A bipartite graph G=(V, E) is an undirected graph whose vertices can be partitioned into two disjoint sets V_1 and $V_2 = V - V_1$ with the properties (i) no two vertices in V_1 are adjacent in G and (ii) no two vertices in V_2 are adjacent in G. Prove that a graph G is bipartite if and only if it contains no cycles of odd length.

5. (22%) Consider the problem of finding a longest consecutive common subsequence: if A = "abcdefg" and B = "abxcdexfg", then the longest consecutive common subsequence will be "cde" (not "abcdefg"). Show how to solve this problem using dynamic programming. Your answer should include recurrence relations, boundary conditions, and algorithms.

6. (10%) Why do we need to use a tree structure in computer computation? Give ten most important reasons that you can think of.