PART I:

ALGORITHM

1. (14%) Write a ***quicksort*** so that elements of the array need not be moved.

2. (18%) True or False (defend your answer): the following nondeterministic algorithm solves the No Partion Problem in nondeterministic polynomial time.

```
HalfSet:={};
p:=n;
repeat
    count:=choose({0,1})
    if count=1 then
        HalfSet:= HalfSet∪{p}
    endif
    p:=p-1;
until p=0
if
```

$$\sum_{i \in HalfSet} c_i \neq \sum_{i \notin HalfSet} c_i$$

```
then
    success, print("yes")
else
    failure, print("no")
endif
```

3. (18%) Let $A = (a_1, a_2, \ldots\ldots, a_n)$ and $B = (b_1, b_2, \ldots\ldots, b_m)$ 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_is and b_is are integers. Write an algorithm to determine whether A and B are equal. your algorithm should work in O(n+m) time.

PART II: Data Structure

1.(18%) Assume that we have a file of records to be stored in disk and we want to build index on the file. The index we have chosen is the B-tree. We have the following given information.

Each record of the file is 100 bytes.
The disk is organized into pages and each disk page has a capacity of 4 Kbytes.
The field of each record on which the index is built contains a 4 byte integer.
Each index pointer is 4 byte wide.

Please answer the following questions:

(a) If each node of the B-tree is also 4 Kbytes in capacity, how many leaf nodes will a three-level B-tree have? (The three levels include the root node.)
(b) What is the minimum size of the main memory if the entire B-tree is to be retained in the main memory?
(c) What is the maximum size of a file in records can such a three-level B-tree index? (Assume that the file is in disk.)

2.(20%)

(a) The *radius* of a tree is the maximum distance between any two vertices. Given a connected and undirected graph, write an algorithm for finding a spanning tree of minimum radius.
(b) Analyze the complexity of your algorithm.

3.(12%) Let P be a pointer to a unidirectional circularly linked list. Write an algorithm to show how to add elements to and delete elements from the list.