1. (10 points) Which of the following is *false* about branch and bound? **Circle** the false statement (there is only one). You do not need to explain your answer.

   a) Branch-and-bound always gives a correct answer

   b) Branch-and-bound always runs in polynomial time

   c) Branch-and-bound can be used to solve problems other than ILP/MIPs.

2. (10 points) **What is the suffix array** of the following string?

   `baba$`

   Recall that the $ character is considered to be before any other in sorted order. (You probably do not want to use an efficient algorithm to find the suffix array—$O(n^2 \log n)$ is fine.)

3. (10 points) Which of the following is the correct performance guarantee of a van Emde Boas tree (vEB tree) containing n items from a universe of possible items U?

   a) The vEB tree can answer predecessor or successor queries in $O(\log n)$ time

   b) The vEB tree can answer predecessor or successor queries in $O(\log \log U)$ time

   c) The vEB tree can answer membership queries in $O(\log n)$ time

   d) The vEB tree can answer membership queries in $O(\log \log U)$ time

4. (20 points) You have a problem that is *nearly* an LP:

   Objective:

$$\max x + y$$

   Constraints:

$$\begin{aligned} x^2 + y^2 &\leq 9 \\ x - y &\leq 2 \\ x &\geq 0 \\ y &\geq 0 \end{aligned}$$

   Will the simplex algorithm give a correct solution to this problem? Please explain briefly why it always will, or why it may not. Note that I am looking for a reason why simplex will work or fail for *this problem in particular*, not just a statement that it may not work because the problem is not an LP.

   Hint: Drawing a picture of the search space may help.

5. (20 points) The following is a variant of the Travelling Salesman Problem.

   Let's say you have a collection of n points; for each pair of points i and j there is a cost c_{ij} to travel from i to j.

   In addition, k ***target points*** are identified (they are a subset of the n points above).

   Your goal is to find the shortest cycle that visits all k of the target points. Your cycle may also visit points that are not target points—the only requirement is that at least the k target points *must* be visited. To be clear, the solution must be a "cycle": the cycle must start and stop in the same place, and the cycle cannot visit any point more than once. (In particular, each of the target points must be visited *exactly* once; each of the $n - k$ points that are not target points can be visited *at most* once.)

   The input consists of: n, k, the n^2 values for c_{ij}, and a list of k target points.

   Give an MIP or ILP for this problem. Its size and construction time should be polynomial in n. **You do not need to prove that it is correct or bound the size or construction time** (though the proof may be a good exercise to verify your solution).

6. (20 points) Consider the string of length $3k + 1$:

$$\mathtt{c}^k\mathtt{b}^k\mathtt{a}^k\mathtt{\$}$$

(that is to say: the letter c k times, followed by the letter b k times, followed by the letter a k times, then the dollar sign). What is the Burrows-Wheeler Transform of this string? Your answer should be in terms of k (that is to say: you should give the correct answer for any k).

7. (15 points) You want to run branch and bound on the following knapsack instance with five items, with total weight constraint 8. (Recall that we want to find the subset of items with the maximum value that have total weight at most 8.)

Item	**Value**	**Weight**
1	40	2
2	13	3
3	12	2
4	13	4
5	11	3

Your branch and bound method splits the state space into three partitions:

1. The first partition contains all solutions that contain the first item, but not the second
2. The second partition contains all solutions that contain the second item, but not the first
3. The third partition contains all solutions that contain both the first and second item.

After the branch-and-bound method finds the best solution in the first partition, it only needs to search one of the second and third partition; the other can be skipped. Explain which one can be skipped and why.