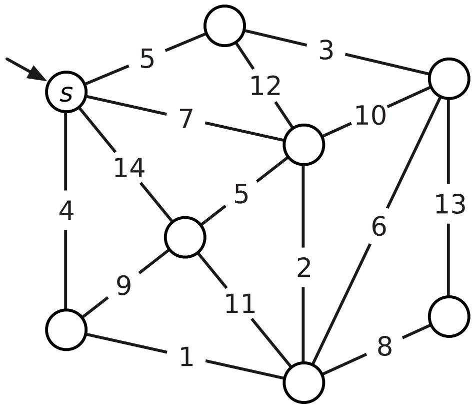
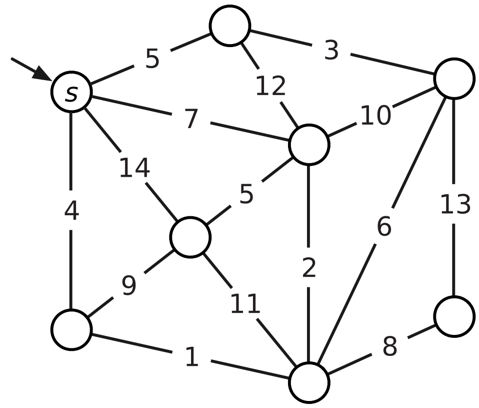
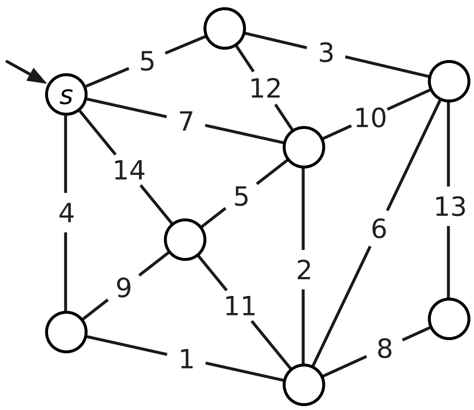
1. **[10 points]** Clearly indicate the four indicated spanning trees of the same undirected edge-weighted graph.



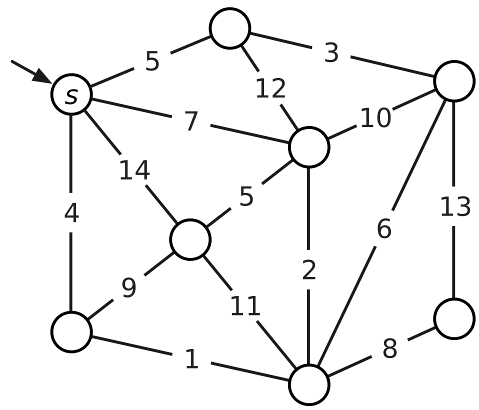
(a) A breadth-first search tree rooted at *s*.



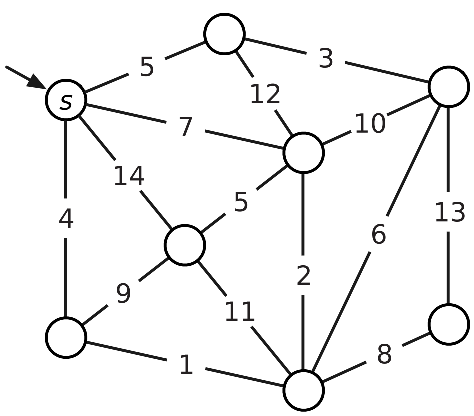
(b) A shortest-path tree rooted at *s*.

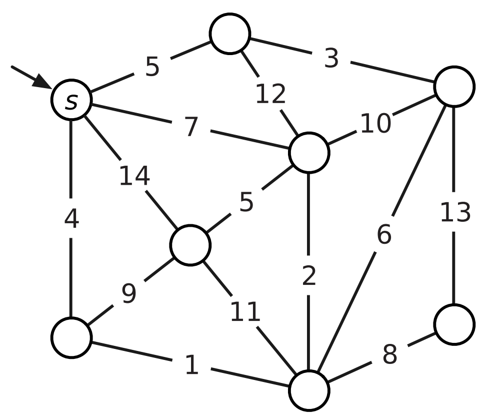


(c) The minimum spanning tree.



(d) The *maximum* spanning tree.





(extra copies for scratch work)

2. **[15 points]** Let L be the set of strings $\{0^n 1^{2n} \mid n \geq 0\}$. For example, the strings 001111 and 000011111111 are in L, but the strings 101110 and 00111 are not.

   (a) Describe a context-free grammar for L.

   (b) Describe a context-free grammar for the complement $\{0,1\}^* \setminus L$.

   In both grammars, give a *brief* description of the language generated by each non-terminal.

3. **[20 points]** Recall that a *palindrome* is any string that is the same as its reversal. For example, `I`, `DAD`, `HANNAH`, `AIBOHPHOBIA` (fear of palindromes), and the empty string are all palindromes.

   (a) Describe and analyze an algorithm to find the length of the longest *substring* (not *subsequence*!) of a given input string that is a palindrome.

   For example, **`BASEESAB`** is the longest palindrome substring of `BUB`**`BASEESAB`**`ANANA` ("Bubba sees a banana."). Thus, given the input string `BUBBASEESABANANA`, your algorithm should return the integer 8.

   (b) Any string can be decomposed into a sequence of palindrome substrings. For example, the string `BUBBASEESABANANA` can be broken into palindromes in the following ways (and many others):

```
BUB + BASEESAB + ANANA
B + U + BB + A + SEES + ABA + NAN + A
B + U + BB + A + SEES + A + B + ANANA
B + U + B + B + A + S + E + E + S + A + B + A + N + A + N + A
```

   Describe and analyze an algorithm to find the smallest number of palindromes that make up a given input string. For example, given the input string `BUBBASEESABANANA`, your algorithm should return the integer 3.

(additional space for problem 3)

4. **[15 points]** Binaria uses coins whose values are $1, 2, 4, \ldots, 2^k$, the first k powers of two, for some integer k. As in most countries, Binarian shopkeepers always make change using the following greedy algorithm:

```
MakeChange(N):
  if N = 0
      say "Thank you, come again!"
  else
      c ← largest coin value such that c ≤ N
      give the customer one c¢ coin
      MakeChange(N − c)
```

For example, to make 37¢ in change, the shopkeeper would give the customer a 32¢ coin, a 4¢ coin, and a 1¢ coin, and then say "Thank you, come again!" (For purposes of this problem, assume that every shopkeeper has an unlimited supply of each type of coin.)

Prove that this greedy algorithm *always* uses the smallest possible number of coins.

5. **[10 points]** You just discovered your best friend from elementary school on Twitbook. You both want to meet as soon as possible, but you live in two different cites that are far apart. To minimize travel time, you agree to meet at an intermediate city, and then you simultaneously hop in your cars and start driving toward each other. But where *exactly* should you meet?

   You are given a weighted graph $G = (V, E)$, where the vertices V represent cities and the edges E represent roads that directly connect cities. Each edge e has a weight $w(e)$ equal to the time required to travel between the two cities. You are also given a vertex p, representing your starting location, and a vertex q, representing your friend's starting location.

   Describe and analyze an algorithm to find the target vertex t that allows you and your friend to meet as quickly as possible.

(scratch paper)

(scratch paper)