## Section A. Answer at least ONE question

1. a. i. What is the heap property?
ii. Why is a heap with the heap property holding a useful data structure?
iii. Give an example of a heap with the heap property holding containing 8 nodes.
[6 marks]
b. i. Briefly explain how Huffman coding works
ii. Is the Huffman coding algorithm a greedy algorithm, a dynamic programming algorithm, or a divide and conquer algorithm. Why?
iii. In what sense does Huffman coding produce an optimal encoding? Give an equation to clarify.
[12 marks]
c. Give an outline proof that the Huffman coding algorithm produces an optimal encodation.
2. See 2b12.q2.ps
3. See 2b12.q3.ps

## Section B. Answer at least ONE question

4. a. Give a formal definition of a finite state machine (deterministic), explaining when such a machine accepts its input and when it does not.

Consider the finite state machine $M$ with states $\left\{q_{0}, q_{1}, q_{2}\right\}$, initial state $q_{0}$, halting states $\left\{q_{2}\right\}$, alphabet $\{a, b, c\}$ and transition function $\delta$ defined by:

| state | symbol | new state |
| :--- | :--- | :--- |
| $q_{0}$ | $a$ | $q_{0}$ |
| $q_{0}$ | $b$ | $q_{1}$ |
| $q_{0}$ | $c$ | $q_{2}$ |
| $q_{1}$ | $a$ | $q_{2}$ |
| $q_{1}$ | $b$ | $q_{0}$ |
| $q_{2}$ | $c$ | $q_{2}$ |
| $q_{2}$ | $a$ | $q_{1}$ |

b. Draw a state-transition diagram to represent the machine $M$.
[5 marks]
c. For each of the following strings, say what $M$ would do if the string was used as the input for a computation.
i. $a a c c b$
ii. $b b c c$
iii. $b a b$
iv. $a b b c a$
[8 marks]
d. Give a formal definition of a non-deterministic finite state machine, explaining when it accepts its input and when it does not.
e. Design a non-deterministic finite state machine for the alphabet $\{a, b\}$ so that the machine accepts its input if and only if the input string contains a substring $a b b \ldots b a$ where the number of $b s$ in the substring is a multiple of three. You may either draw the state-transition table or write down the transition function.
5. Consider a queue for a printer where print jobs arrive at certain times and are added to the end of the queue and where the print job at the front of the queue can be de-queued and printed. Let the flow of time be the natural numbers $\mathbb{N}$. Let $Q^{2}, P^{2}$ be binary predicate symbols and let $Q^{2}(p, t)$ denote that the print job $p$ is added to the queue at time $t$ and $P^{2}(p, t)$ denotes that $p$ is being printed at time $t$. Let $<^{2}$ be another binary predicate symbol and let $<^{2}\left(t_{1}, t_{2}\right)$ denote that the time $t_{1}$ is earlier than the time $t_{2}$ (you can also use the infix notation $t_{1}<t_{2}$ if you prefer). For each natural number $n \in \mathbb{N}$ let $n$ be a constant symbol denoting itself.
a. For each of the following sentences, write down an equivalent formula in first-order logic.
i. "If one print job arrives before another then it is printed first."
ii. " At time 7 nothing will be being printed."
iii. "Any print job that arrives at the queue will eventually be printed."
b. Translate these formulas into English.
i. $\forall x \neg \exists t\left(t<13 \wedge P^{2}(x, t)\right)$
ii. $\forall t\left(t<50 \rightarrow \exists x P^{2}(x, t)\right)$
iii. $\exists x \forall t P^{2}(x, t)$
iv. $\forall t \exists x P^{2}(x, t)$
v. $\forall t \forall x\left(P^{2}(x, t) \rightarrow n<t\right)$.
c. State the compactness theorem for first-order logic.
d. Prove that the theory

$$
\left\{\forall t \forall x\left(P^{2}(x, t) \rightarrow n<t\right): n \in \mathbb{N}\right\} \cup\left\{\exists t \exists x P^{2}(x, t)\right\}
$$

has a model.

END OF PAPER

