If you lost x marks for something, there should be a $-x$ next to the comment. For common mistakes, I've tried to correlate comments written on the paper to the italicized terms in this marking guide, to give a more complete explanation of the mistake.

Question 1

5 marks

- 1 mark: *Justify* $\alpha/1-\alpha$ *split:* the recursive split is between α and $1-\alpha$, but not necessarily with equality. You should justify that to get a minimum or maximum height recursion tree, we will get a split of α and $1-\alpha$ at each step.
- 2 marks: show the minimum. 1 mark for justifying $\alpha^i n \leq 1$, one mark for solving this.
- 2 marks: show the maximum.
- *Need something for every choice of* α: Some people chose a single value of α. Your answer needs to work for every value of α. -3 marks for this.

Question 2

10 marks

- 5 marks: Algorithm
  - 3 marks: hash the terms x_i and/or $w_j = z-y_j$, compare the elements in the hash tables to check for equal values
  - 1 mark: *specify hash table size*
  - 1 mark: *how do you handle collisions?*
  - $O(n^2)$ algorithm: not more than 2 marks total.
  - 1 mark off for using a specific universal hash family that was not actually a universal hash family.
- 5 marks: Time Analysis:
  - 1 mark: show each loop is size $O(n)$.
  - 4 marks: *Justify* $O(1)$ *with defn of universal hashing*: You should use the definition of a universal hash family (or invoke a theorem) to show that the chains in the hash table are a constant length. Equivalently, show lookups take $O(1)$ time. This implies that the inside of each loop takes only $O(1)$ time.
    - Good analysis for a single hash function h (no universal family) is worth 2 marks.

* Showing Pr(collision) $= \frac{1}{n}$ but not showing this implies expected insert/lookup time is $O(1)$ is worth 2 marks.

- 2 bonus marks: Do the above with a specific universal hashing scheme properly (probably one from class/textbook)

Question 3

a) 4 marks

- 3 marks: *lower bound*
  - *want worst case lower bound*: In this context, finding a lower bound means finding a specific sequence
- 1 mark: *upper bound:*
  - In this context, finding a upper bound on the worst case means showing no sequence of operations can take longer than $O(m)$.

b) 6 marks:

- 2 marks: *give credit scheme*:
  - there was a lot of flexibility given in what counted as an "operation" (and therefore what a valid credit scheme could be) but ENQUEUE should be one operation; for DEQUEUE, inside body of the loop should be one operation, and the pop from H should be one operation.
  - If you charge 0 for DEQUEUE, then your scheme will fail for a sequence of dequeues on an empty stack. No marks off for this.
- 2 marks: *give a credit invariant*: it was ok if you stated this informally.
- 2 marks: *prove credit invariant*
  - If you have a faulty credit scheme/invariant, it was still possible to get these marks.

Question 4

a) 6 marks

- 2 marks: recognize amortized analysis and choose an appropriate method (accounting method, others also work) to approach the problem
- 2 marks: recognize that for an array of size n, increasing from size $\frac{2n}{3}+1$ to size $n+1$ takes $O(n)$ operations, including the cost for increasing the array to size $2n$ when the last element is inserted.

- 2 marks: give an argument that works.
- Many people didn't recognize that

$$\sum_{i=1}^{\log_{3/2} m} \left(\frac{3}{2}\right)^i \leq m + \frac{2}{3}m + \left(\frac{2}{3}\right)^2 m + ... \leq 3m$$

b) 4 marks

- 2 marks: recognize that it is now $\Omega(m^2)$.
- 2 marks: show it correctly (i.e. give an example)
- *Want lower bound*: Note that an amortized analysis approach for this question doesn't work well. Amortized analysis can prove an upper bound of $O(m^2)$, but this is obvious because even if you grow the array at every insertion, you are only doing m^2 operations. Counting the cost of a specific example of operations gives a lower bound of $\Omega(m^2)$.

Solutions with an asymtotic complexity that was not simplified (ie, involved a summation or involved an extra$m$) lost one mark.