UNIVERSITY *of* LIMERICK

OLLSCOIL LUIMNIGH

COLLEGE *of* INFORMATICS *and* ELECTRONICS

Department of Computer Science
and Information Systems

End-of-Semester Assessment Paper

Academic Year:	1997/98	Semester:	Autumn
Module Title:	Data Structures and Algorithms	Module Code:	CS4115
Duration of Exam:	$2\frac{1}{2}$ hours	Percent of Total Marks:	50
Lecturer:	P. Healy	Paper marked out of:	100

Instructions to Candidates:

- Answer *all* questions
- All questions carry equal marks
- Please keep your answers *precise* and *concise*

Q1. Short questions: 4 marks each (20 marks)

(i) For a *closed hash table*, give a bound on the load factor, λ, for acceptable performance

(ii) What is the worst-case running time for building a *binary heap* on n nodes

(iii) As a function of n, what is the running-time (in Big-Oh notation) of the section of code shown below?

```
sum = 0;
for (int i = 1; i <= n; i++)
  for (j = 1; j <= i*i; j++)
    if (j%i == 0)
      for (int k = 0; k < j; k++)
        sum++;
```

(iv) What is the worst-case running time of Depth First Search on a graph $G = (V, E)$?

(v) What is the expected number of inversions in a list of n numbers? Justify your answer.

Q2. (20 marks)

The *burger* flip takes an object and turns it upside down. The *stack o' burgers* flip (SOB) generalizes the burger flip by turning a stack of objects upside down so that of n objects, the i^{th} from the top will be relocated to the i^{th} from the bottom after the flip.

(i) Provide an algorithm that will sort a stack of n integers using the SOB flip as its basic operation.

(ii) What is the worst-case running time of your algorithm in terms of SOB flips?

(iii) What is the worst-case running time of your algorithm in terms of number of integers moved around?

(iv) Give an input data set that achieves this worst case bound.

Q3. (20 marks)

Given the input 4371, 1323, 6173, 4199, 4344, 9679, 1989 and a hash function $h(x) = x \bmod 10$

(i) With $hsize = 10$, show the resulting tables under

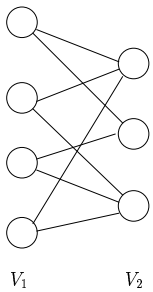
1. open hashing
2. closed hashing using linear probing
3. closed hashing using quadratic probing
4. closed hashing with secondary hash function $h_2(x) = 7 - (x \bmod 7)$

(ii) Show the result of rehashing the tables you got above

Q4. (20 marks)

A *bipartite* graph is a graph, $G = (V, E)$, such that V can be partitioned in to two disjoint sets V_1 and V_2 and no edge has its two endpoints in the same set. A picture of a bipartite graph is shown below.

Prove that a graph is bipartite if and only if it contains no cycles with an odd number of edges.



Q5. (20 marks)

Argue carefully that the code shown below generates a random permutation of the numbers 1 to n in the array, `arr[]`. The function `rand_int(i, j)` returns a random integer in the range $[i, \ldots, j]$.

```
template <class T>
void swap(T& a, T& b)
{
  T tmp = a; a = b; b = tmp;
}

for (int i = 0; i < n; i++)
  arr[i] = i+1;

for (int i = 1; i < n; i++)
  swap(arr[i], arr[rand_int(0, i)]);
```