EECE72 Fundamentals of Computer Engineering Homework 1

Shaoyang Yan(001220697)

1 (a)

$$\sum_{i=1}^{n} i = 1 + 2 + .. + i + .. + (n-1) + n$$

$$\sum_{i=1}^{n} i = 1 + 2 + .. + i + .. + (n-1) + n$$

Sum the two equations above we can get:

$2x\sum_{i=1}^{n} i = (n+1) + (n+1) + .. + (n+1) = n(n+1)$

Then divide the equation above by 2

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

(b)2 steps

Mathematical induction

Base case:

prove the given statement for the first natural number

Second step(inductive step):

prove that the given statement for any one natural number implies the given statement for the next natural number.

It goes wrong when n=1. Of course every set of n(n=1) babies have the same color eyes. But the whole set is n+1=2 babies. No intersection exists between L and R which can be translated into

this format(L∩R=Ø).

2

(a)

$$A \cup B = \{x | x \in A \vee x \in B\}$$

(this means elements are in A or B)

$$A \cup B = \{x | x \in A \wedge x \in B\}$$

(this means elements are in both A and B)

(b)

Let S be a finite set with n elements. What we want to show is that

P(n): S have 2^n subsets.

We start the induction with n=0. This means there is no element in S, thus the only one subset of S is itself, empty set.

Suppose P(k) is true. P(k) have 2^k elements. Let S have k+1 elements and c be one element in S. S-{c}=$\{x \in S | x \neq c\}$. S-{c} has k elements, and S-{c} has 2^k subsets. There are two kinds of subsets of S, one with c and one without c. The kind of elements without c have 2^k subsets. The kind of subsets containing c consists of one of the 2^k subsets not containing c, with c adjoined. The number of subsets of subsets containing c

is also 2^k. To sum up, the total number of subsets of S is:

$$2^k+2^k=2*2^k=2^{k+1}$$

P(k+1) is true. P(n) is true for all nonnegative integers n.

(c) We can find an one-to-one correspondence relationship between odd numbers and natural numbers.

$$\mathrm{n} \leftrightarrow \begin{cases} -(n+1), n\ is\ even \\ n \qquad\qquad , n\ is\ odd \end{cases}$$

Thus, we can say there are as many odd numbers as natural numbers.

3

(a)

g(n)=O(f(n)):

There exits positive constant c and natural number n_0 such that when $n \geq n_0$ $|g(n)| \leq c|f(n)|$.

When f:N→N

(b)

T

T

F

F

T

(4)

(a)

A tree is an undirected graph in which any two vertices are connected by exactly one path. An undirected graph can be judged as a tree, with following criterion.

The undirected graph has no circle

The undirected graph is connected

(b)

If a tree has n vertices, its maximum possible length of a path in this tree is n-1.

5

(a)

Definition:

array: Data structure of related(same data) data items of the same type. Static data structures(remain the same size). saved continuously in memory, easy to be locate by index. Add or delete an element in array is difficult because large number of elements need to be moved.

lists: saved separately in memory, and elements are connected by pointers. Difficult to find elements and easy to

add or delete element.

comparison:

the elements of an array are on continuous locations on memory.

And the memory space of array is allocated before compiling

The elements in a array can be accessed through indexes.

Insertion and deletion are very costly in an array.

The elements in linked list can be stored in separate places.

The memory space of linked list can be allocated during execution of program.

The elements in a linked list can only be accessed by linked nodes.

The deletion and insertion do not cost a lot.

(b)

The concatenated list from two original lists requires the connection of the last node of the head list and the first node of the last list.

Make the address field of a_n point to the location of b_1 in memory.

6

(a)

Insertion sort

```
For j=2 to n
      Key =A[j]
      I=j-1
      While(i>0)&&(A[i]>key)
            A[i+1]=A[i]
            I=i-1
      End while
      A[i+1]=key
   End for
   Return A
End function
```

(b)

Unsorted

```
Min=A[1]
   For i=2 to n
      If(min>A[i])
      Min=A[i]
      End if
   End for
```

Return min

End function

Sorted

If(A[1]<A[n])

Return A[1]

Else

Return A[n]

The searching time changes if an array is sorted. Before sorting, the time cost is O(n), and after sorting the time cost is O(1).

7

(a)

$\sum_{i=1}^{n-1} \mathrm{i} * (\mathrm{i}+1)$ =1*2+2*3+…+n(n-1)=1*(1+1)+2*(2+1)+…+(n-1)(n-1+1)=1^2+…+(n-1)^2+1+…+n-1=n(n-1)(2n-1)/6+n(n-1)/2=n^3/3-n/3

The value returned is n^3/3-n/3

(b)

For there is a triple loop in this function, and each loop is related to n. Thus the time complexity is O(n^3)

8

(a)

A function prototype is a declaration of a function that specifies the function's name and type signature(arity, parameter types, and return type), but omits the function body.

(b)

Pass by value：

The argument's value is copied into the function's parameter.

Pass by address:

Passing an argument by address involves passing the address of the argument variable rather than the argument variable itself. Because the argument is an address, the function parameter must be a pointer. The function can then dereference the pointer to access or change the value being pointed to.

Pass by reference:

To pass a variable by reference, we simply declare the function parameters as references rather as normal variables.

Differences:

Passing by reference means the called functions' parameter will be the same as the callers' passed argument(the variable itself). Passing by value means the called functions' parameters will be a copy of the callers' passed argument. Passing by address means that the argument will be addresses and the function parameters must be pointers.

9

(a) An identifier becomes visible with its declaration. The region where an identifier is visible is referred to as the identifier's scope.

(b) i main function scope

ii function scope

iii global scope

iv filescope

v global scope

vi function prototype scope

10

(a) The first condition (a==b) is considered first. If the first condition evaluates to be true, the second condition will not be evaluated at all.

(b) For OR operator, put the more likely true condition on the left side.

(c) (a==b)=0 (a+b>32)=0

Thus the value of this expression will be 0