CSC165H1: Problem Set 3

Due November 15, 2017 before 10pm

1.（a）

WTS:

P(n):

Proof: We will prove this statement using induction on n.

Let , let

**Base case:**

Let n=1.

We assume that is and , we want to prove that that is

Since (course note 2.18(c))

And and （what we assumed before）

So

Hence

**Induction step:**

Let and assume that

We want to prove that Assume that we want to prove that

Since and n<n+1 so, so (by induction hypothesis)

Since so

Since (course note 2.18(c))

Hence that is

In all, we have proven that

(b)

WTS:

P(n):

Proof: We will prove this statement using induction on n.

Let , let , we assume that

**Base case:**

Let n=0

We assume that , that is

We want to show that that is

We will prove this by contradiction

Assume that d|

Since d|d, so

Hence

So

**Induction step:**

Let

We want to prove that

Assume that

We want to show that

We will prove this by contradiction

Assume that that is

Since and n+1>n

So so , that is gcd()=1 (by induction hypothesis)

Since (2(g) from problem set 2)

Since gcd()=1 and and and

Thus d|

So gcd(d,

But we already know from our assumption that gcd(d, , so we get a contradiction

Hence

In all we have proven that

(c)

WTS:

P(n):

Proof:

We will prove this statement using induction on n.

**Base case:**

Let n=2 (2>1)

So

**Induction step:**

Let

We assume that

We want to prove that

Since 2n+2>2n+1>0

So

So

So (by induction hypothesis)

Hence

In all we have proven that

(d)

WTS:n∈

P(n):

Proof: We will prove this statement using induction on n.

Let

**Base case:**

Let n=0, ,

**Induction step:**

Let n∈,

Assume

We want to show that

Since n∈, so n0, hence n+1>0

Thus:

= (by induction hypothesis)

=

=

In all, we have proven that n∈

2.(a)

WTS:

P(n):

Proof:

, assume k=n, we know that the number of subsets S of size |s| is always 1, so and , hence is always true.

, assume k=0, we know that the number of empty subset of a set is always 1,

so and , hence is always true.

Therefore, we only need to prove

We will prove this statement using induction on n.

**Base case:**

Let n=0

Let , assume

Let n=1

Let , assume , that is 0<k<1 since , so this assumption is false,

Let n=2

Let , assume , that is 0<k<2, so k=1, since there are two elements in a set which size is two. So the number of subsets S of size 1 is 2, so

And

Thus . Hence

**Induction step:**

Let , assume that

Since is always true and is always true. We can write the assumption as

We want to show that

Let , we want to show that

When it comes to a set S with |S|=n+1, we can let the elements in the set be Let S’= so that S=S’ |S’|=n

* First, counting subset of size k that contains :

Since every subset of S of size k that contains must contain exactly k-1 elements from S’, there are choices of elements from S’.

Since so, Hence (by induction hypothesis)

Hence there is subsets of S of size k that contains

* Second, counting subset of size k that does not contain :

Every subset of size k of S that does not contain must contain k of the elements . That is, these subsets are exactly the subsets of size k of S’, so the number of these subsets is .

Since so thus ,Hence (by induction hypothesis)

Hence there is subsets of S of size k that does not contain

By combining the two counts from the first and second part, the total number of subsets of size k of S is

That is

Hence we have shown that

In all we have proven that And and

Therefore we have proven that

(b)

The elements of

The elements of

(c)

Proof:

WTS:

P(n):

1. Let n=0

Since we got from the question that

So our statement is true when n=0.

1. We will prove the following statement using induction on n.

**Base case**:

Let n=1

Since we got from the question that

Since

So

**Induction step**:

Let , assume , we assume

We want to prove that

Since , = so that = (Definition 3)

According to the definition of : =

Let , so must be in and only be in one of A,B

If , take C=A without n+1, D=B.

If , take C=A, D=B without n+1.

Let’s first consider the number of {C,D}, then consider the number of {A,B} which is

Since definition is : and we also know that =

So the definition of is : and and

Hence we can find that , so the number of is equal to which is (by induction hypothesis)

Now let’s consider the number of {A,B} which is , since

If , take C=A without n+1, D=B.

If , take C=A, D=B without n+1.

{A,B} is add up with n+1, for every , we can add up n+1 on C or on D, so there are two ways to add on n+1. That is the number of {A,B} is equal to . Hence

In all we have proven that

3.（a）

WTS:Theorem 5.8.: For all , if f(n) is eventually greater than or equal to 1, then and

The definition of big-Theta: Definition 5.6. Let . We say that g is (Big-)Theta of f if and only if g is both Big-Oh of f and Omega of f. In this case, we can write

Equivalenty, g is Theta of f if and only if there exist constants such that for all

Proof:

Let we assume that f(n) is eventually greater than or equal to 1 that is , let be that value.

We want to show that

And

First, from the worksheet, we know that: Given any real number x, the floor of x, denoted, is defined to be the largest integer that is less than or equal to x. Similarly, the ceiling of x , denoted , is defined to be the smallest integer that is greater than or equal to x. Hence we know that and

* show that

1. If f(n) is eventually greater than or equal to 1 but smaller than 2:

That is , let be that value.

Let , so , since

Let , assume

We want to show that

* Since , so , so

So

Since , so , so

Hence

That is

* Since , so , so

So (since )

Hence we have shown that

1. If f(n) is eventually greater than or equal to 2:

That is , let be that value.

Let , so , since

Let , assume

We want to show that

* Since , so , thus

So

Thus

So

Since no matter what f is

So

* Since no matter what f is

So (since )

Hence we have shown that

In all we have shown that

* Show that

Let , so , since

Let , assume

We want to show that

* Since no matter what f is

So (Since )

* Since , so , so

So

Thus (Since )

We also know that no matter what f is

Hence

Hence we have shown that

In all we have shown that

Therefore we have proven that For all , if f(n) is eventually greater than or equal to 1, then and

(b)

WTS：

That is:

Proof:

Let ,assume

We want to prove that

Let , take , so

We want to prove that

* We want to prove that :

Since

So

Hence

* We want to prove that :

Since

So

Hence

Since so

Hence

That is

Since so

Hence

Hence we have shown that

Therefore we have proven that

(c)

WTS：

Proof:

Let , we want to show that

( There is one assignment and one return sentence in the whole function and we will consider them only when there is no iteration in the loop. Cause (take c=1 and =1000, so ) and from therem5.5 we know that for all , so we only need to consider whether the running time of the loop is a big-Oh of .)

assume

* CASE1: n=m:

If n=m , then the quotient will be r0//r1=1

Hence after one iteration, r1=n-(n//m)m=n-m=0

And the loop is over.

Hence when n=m, (take c=1 and =1000, so )

* CASE2: m<n:

Let be the original value of r0, be the value of r0 after one loop iteration, and the value of r0 after two loop iterations. We want to prove that

From the question, we get that

Since m<n from the “r0, r1 = r1, r0 - quotient \* r1”, we know that , since is either be the remainder that the last r0 divides or be m when be n.

We divide up this proof into two cases:

* Case1:

(If , although the loop will end at that time, we may assume that the loop is going on till the next iteration when is something related with and the loop end at that time. Since we are calculating the worst situation.)

Since

So

Since and

Thus , that is

* Case2:

Since , so (since if . for For , then , we get a contradiction. Thus )

So

Since

So

So (since )

Hence

Thus , that is

In all we have shown that

That is every two iterations of the loop reduces r0 by at least half.

And the loop will be over when r1=0 that is when r0 becomes the gcd(n.m), (since the loop is to get the extended gcd(n,m) and r0=gcd(n,m), r0=s0n+t0m)

Since we want to get the big-Oh of , hence we can think about the worst condition that is gcd(n.m)=1. Hence when , the loop will get over. (Since sometime we won’t get 1 when divide n by 2 everytime and the gcd(n,m) must be greater or equal to 1)

After 2k iterations, r0 will be , and take this into , we get , so

Hence the number of actual iterations is at most , with each iteration costing a single step

Thus

Take so

, hence

* CASE3: m>n:

So the first iteration will do the work that exchange the value of r1 and r0, so r1 will be n and r0 will be m, and the second iteration will do the work that change n back to r0, and r1 at this time is the remainder of m divides n which is smaller than n, let’s call it m’.

**Start with the third iteration.** Let be the original value of r0, be the value of r0 after one loop iteration, and the value of r0 after two loop iterations. We want to prove that

From the question, we get that

Since m’<n from the “r0, r1 = r1, r0 - quotient \* r1”, we know that , since is either be the remainder that the last r0 divides or be m’ when be n.

We divide up this proof into two cases:

* Case1:

(If , although the loop will end at that time, we may assume that the loop is going on till the next iteration when is something related with and the loop end at that time. Since we are calculating the worst situation.)

Since

So

Since and

Thus , that is

* Case2:

Since , so (since if . for For , then , we get a contradiction. Thus )

So

Since

So

So (since )

Hence

Thus , that is

In all we have shown that

That is every two iterations of the loop reduces r0 by at least half.

And the loop will be over when r1=0 that is when r0 becomes the gcd(n.m), (since the loop is to get the extended gcd(n,m) and r0=gcd(n,m), r0=s0n+t0m)

Since we want to get the big-Oh of , hence we can think about the worst condition that is gcd(n.m)=1. Hence when , the loop will get over. (Since sometime we won’t get 1 when divide n by 2 everytime and the gcd(n,m) must be greater or equal to 1)

After 2k iterations, r0 will be , and take this into , we get , so

Hence the number of actual iterations is at most (since the first iteration will do the work that exchange the value of r1 and r0, so r1 will be n and r0 will be m, and the second iteration will do the work that change n back to r0, and r1 at this time is the remainder of m divides n which is smaller than n, so we gonna add 2 here) ,with each iteration costing a single step

Thus

Take so

, hence

If m=0 :

So the condition is wrong at the first time

And (take c=1 and =1000, so )

If n=0 and m

There is only 1 iteration

And (take c=1 and =1000, so )

Therefore we have shown that