**Algorithmics Unit 4 Week 1 – Submission Task**

1. Evaluate:

5+9+13=33 [if n=i]

Or 4 x (4n+1) = 12n+3 [if n is independent of i]

6+12+20+30 [if n=i]

Or 5 x n(n+1) = 5n^2+5n

1. If find in terms of :

= 4k

= k+50

1. Prove by induction (for all

Base case:

For n=1 -> 2^1-1 = 1

Inductive step:

Assume it holds true for n=k

+ ... + 2^(k-1) = 2^k - 1.

Prove it holds true for n=k+1

... + 2^k + 2^(k+1) = 2^(k+1) – 1

Proof:

1. (1 + 2 + 4 + 8 + ... + 2^k) + 2^(k+1)
2. Sub 2^(k+1)-1
3. LHS = (2^(k+1) - 1) + 2^(k+1)
4. RHS = 2^(k+1) – 1
5. (2^(k+1) - 1) + 2^(k+1) = 2 \* 2^(k+1) – 1
6. 2 \* 2^(k+1) is the same as 2^(k+2)
7. Therefore
8. Consider the following algorithm, which searches for ‘x’ in array ‘A’ and returns either the index (if A contains x) or -1 if A does not contain x:

def search(A,x):

n = length of array

for i = 1 to n:

if A[i] = x then return i

return -1

(Here indices start at 1, and A[2:4] includes A[2],A[3] and A[4].)

1. Which of the following are loop invariants?
2. At the start of each iteration, if A contains x then the subarray A[i:n] contains x.
3. At the start of each iteration, if A contains x then the subarray A[i+1:n] contains x.
4. At the start of each iteration, A[1:i] does not contain x.
5. At the start of each iteration, A[1:i-1] does not contain x.
6. At the start of each iteration, if A[i:n] does not contain x then x is not in A.
7. At the start of each iteration, if A[i+1:n] does not contain x then x is not in A.
8. At the start of each iteration, if A[i-1:n] does not contain x then x is not in A.

(b) For the first loop invariant you have identified, explain whether or not it remains true after the loop has completed.