Question 1

```
maxSum(A, j):
    if(j == 0):
        return A[0]
    prev_max = maxSum(A, j - 1) + A[j]
    return max(prev_max, A[j])
```

Question 2

(a)

```
maxSumMemo(A, j T):
    if(j == 0):
        return A[0]
    if(T[j] != -infinity):
        return T[j]
    prev_max = maxSumMemo(A, j - 1, T) + A[j]
    T[j] = max(prev_max, A[j])
    return T[j]
```

```
maxSum(A):
    n = A.length
    T is a table/array of length n, whose values are all initialised to -infinity
    max = maxSumMemo(A, n - 1, T)
    for(k = n - 2; k >= 0; k--):
        curr = maxSumMemo(A, k, T)
        if(curr > max):
            max = curr
    return max
```

(b)

In maxSum(), when maxSumMemo() is called for the first time outside the for-loop, it recursively iterates through all n entries of A, filling up the table T with their maximum subarray sums. This takes n steps.

Then in maxSum(), the for-loop iterates approximately n times, calling maxSumMemo() each time. However, this time maxSumMemo() will only take a fixed number of steps, since the table T has already been filled out. So, the for-loop takes n steps.

So, the number of steps is approximately n + n. Which is a $\Theta(n)$ cost.

Question 3

(a)

```
maxSumTabulate(A):
```

```
n = A.length
T is a table/array of length n
max = T[0]
for(k = 1; k < n; k++):
    T[k] = max(T[k - 1] + A[k], A[k])
    if(T[k] > max):
        max = T[k]
return max
```

(b)

There is one loop that iterates over all n entries, and the cost within the loop is fixed. So, the cost is $\Theta(n)$.

Question 4

(a)

The probability that bad(a, b) returns the correct answer can be split into two scenarios. Particularly, notice that weightedCoin(a) has a $\frac{a-1}{a}$ probability of returning 0 and a $\frac{1}{a}$ probability of returning 1.

Case 1: When weightedCoin(a) returns 0, the correct answer $a + b$ is returned.

Case 2: When weightedCoin(a) returns 1, random(b) is called, with a $\frac{1}{b}$ probability of returning b, in which case the function would return $a + b$, the correct answer.

So, the overall probability that bad(a, b) returns the correct answer is the probability of case 1 occuring plus the probability that case 2 occurs AND returns the correct answer. Thus, the probability that bad(a, b) returns the correct answer is

$$\frac{a-1}{a} + (\frac{1}{a})(\frac{1}{b})$$

(b)

In case 1, the function can only return $a + b$. In case 2, the function and return $a + 1, a + 2, ..., a + b$, each with a probability of $\frac{1}{b}$. So from this information, and the previously found probability of returing $a + b$, we derive the following probability distrubution:

X	a+1	a+2	...	a + b - 1	a + b
P(X)	$(\frac{1}{a})(\frac{1}{b})$	$(\frac{1}{a})(\frac{1}{b})$	...	$(\frac{1}{a})(\frac{1}{b})$	$\frac{a-1}{a} + (\frac{1}{a})(\frac{1}{b})$

Now, we can calucalte the expected value of X, $E(X)$.

$$E(X) = \sum_{k=1}^{b-1}((a+k)(\frac{1}{a})(\frac{1}{b})) + (a+b)(\frac{a-1}{a} + (\frac{1}{a})(\frac{1}{b}))$$

$$= (\frac{1}{a})(\frac{1}{b})\sum_{k=1}^{b}(a+k) + (a+b)(\frac{a-1}{a})$$

$$= (\frac{1}{a})(\frac{1}{b})(ab + \frac{b(b+1)}{2}) + (a-1) + (\frac{b(a-1)}{a})$$

$$= 1 + \frac{b+1}{2a} + a - 1 + \frac{2b(a-1)}{2a}$$

$$= \frac{b+1+2ba-2b}{2a} + a$$

$$= \frac{2ba-b+1}{2a} + a$$

$$= a + b - \frac{b}{2a} + \frac{1}{2a}$$