Analysis of Algorithms

Dynamic Programming

If more than one question appears correct, choose the more specific answer, unless otherwise instructed.

Concept: *memoization*

Assume zero-based indexing.

1. Consider memoizing this function:

```
function f(x)
    {
    if (x == 0)
        return 0;
    else if (x == 1)
        return 1;
    else
        return f(x-1) + f(x-2);
    }
```

   T or **F**: Even though there is only one formal parameter, you will need a two dimensional memoization table because you call recursively call the function twice.

2. **T** or **F**: With zero-based indexing, the size of a table dimension is always one more than the largest value used to index into that dimension.

3. **T** or **F**: Continuing with the previous question, the memoization table's size has to be the original value of x.

4. **T** or **F**: Continuing with the previous question, the memoization table's size has to be $x \times x$, using the original value of x.

5. **T** or **F**: If the formal parameter is getting smaller in the recursive calls, the table will be filled out from smaller to larger indices.

6. **T** or **F**: If the formal parameter is getting larger in the recursive calls, the table will be filled out from smaller to larger indices.

7. **T** or **F**: The number of formal parameters in the function is always equal to the dimension of the dynamic programming table.

8. **T** or **F**: It is necessary to initialize a dynamic programming table with base case values.

9. **T** or **F**: In converting a recursive function to a dynamic programming solution, base cases can be moved to a table lookup function.

10. Consider memoizing this function:

```
function f(x)
    {
    if (x == 0) return 0;
    if (x == 1) return 1;
    return f(x-2) + f(x-1);
    }
```

    What would be memoization table's largest index/indices, where x refers to the original value of x?

    (A) $x + 1$ and x - 1
    (B) x - 2
    (C) $x + 1$
    (D) x
    (E) x - 1
    (F) $x + 1$ and $x + 1$
    (G) x and x
    (H) $x + 1$ and x

11. Consider memoizing this function:

```
function g(x,items,y)
    {
    if (x == 0) return 1;
    if (x < 0) return 0;
    if (y == items.size) return 0;
    return minimum(g(x-items[y],items,y),g(x,items,y+1));
    }
```

Assuming the smallest memoization table possible, What would be the memoization table's largest index/indices, where x refers to the original value of x?

(A) x - 2
(B) x - 1
(C) $x + 1$
(D) x
(E) $x + 1$ and *items.size* + 1
(F) x and *items.size - 1*
(G) x and *items.size* + 1
(H) x and *items.size*

Dynamic programming

12. Consider using dynamic programming to improve the efficiency of the following function:

```
function f(a,b,c,d,e)
    {
    if (a == 0) return 0;
    if (a < 0) return -INFINITY;
    if (d == e) return 0;
    return
        max(
           f(a,b,c,d+1,e),
           f(a-b[d],b,c,d,e) + c[d]
           );
    }
```

What would be the dimensionality of the dynamic programming table?

(A) 4
(B) 6
(C) 2
(D) 3
(E) 5
(F) 1

13. Consider using dynamic programming to improve the efficiency of the following function:

```
function f(a,b,c,d,e)
    {
    if (a == 0) return 0;
    if (a < 0) return -INFINITY;
    if (d == e) return 0;
    return
        max(
           f(a,b,c,d+1,e),
           f(a-b[d],b,c,d,e) + c[d]
           );
    }
```

How would the dynamic programming table be filled, using a as an index?

(A) larger a to smaller a
(B) a is not used as an index
(C) smaller a to larger a

14. Consider using dynamic programming to improve the efficiency of the following function:

```
function f(a,b,c,d,e)
    {
    if (a == 0) return 0;
    if (a < 0) return -INFINITY;
    if (d == e) return 0;
    return
        max(
           f(a,b,c,d+1,e),
           f(a-b[d],b,c,d,e) + c[d]
           );
    }
```

How would the dynamic programming table be filled, using b as an index?

(A) larger b to smaller b
(B) smaller b to larger b
(C) b is not used as an index

15. Consider using dynamic programming to improve the efficiency of the following function:

```
function f(a,b,c,d,e)
    {
    if (a == 0) return 0;
    if (a < 0) return -INFINITY;
    if (d == e) return 0;
    return
        max(
           f(a,b,c,d+1,e),
           f(a-b[d],b,c,d,e) + c[d]
           );
    }
```

How would the dynamic programming table be filled, using d as an index?

(A) d is not used as an index
(B) smaller d to larger d
(C) larger d to smaller d

16. Consider using dynamic programming to improve the efficiency of the following function:

```
function f(a,b,c,d,e)
    {
    if (a == 0) return 0;
    if (a < 0) return -INFINITY;
    if (d == e) return 0;
    return
        max(
           f(a,b,c,d+1,e),
           f(a-b[d],b,c,d,e) + c[d]
           );
    }
```

What is wrong, if anything, about the following loop for filling out the dynamic programming table?

```
for (a = 0; a < max_a; ++a)
    for (d = 0; d < max_d; ++d)
       t[a][d] = max(getTable(a,d+1,e),getTable(a-b[d],d,e) + c[d]);
```

(A) none of the other answers are correct
(B) the d loop goes in the wrong direction
(C) the table is filled out correctly
(D) the a loop goes in the wrong direction
(E) one or more of the loop indices is incorrect
(F) there should be three nested loops
(G) there should only be one loop (no nesting)

17. Consider dynamically programming the following function, what should the dimensionality of the table be?

```
!!change this to an actual problem
function f(a,b,c,d,e)
    {
    if(a == e)
        return c/d;
    else if(b < d)
        return 0;
    return f(a,log(2^b),c%8,d + 1,e - 1)
    }
```

(A) None of these answers are correct
(B) 4
(C) 0
(D) 3
(E) 1
(F) 2

For the following questions, consider dynamically programming a function f to implement the *dice rolling* algorithm, which calculates total number of ways to reach a sum z, rolling x dice, with each die having y sides (sides are number from 1 to y). So if you have five dice each with six sides and desire to find the number of ways to achieve a sum of 1, then `f(1,5,6)` should return 0.

18. What is the dimesionality of the table?

(A) 3
(B) 4
(C) none of these answers are correct
(D) 2
(E) 1
(F) 0

19. What is/are the parameter/parameters that fill the entries of the table?

(A) y only
(B) z only
(C) x and y
(D) x, y, and z
(E) none of these answers are correct
(F) z and y
(G) x only
(H) z and x

20. What is/are the largest index/indices into the dynamic programming table?

(A) none of these answers are correct
(B) x - 1, y - 1
(C) $x + 1$ and z
(D) $x + 1$ and $y + 1$
(E) $x + 1$, $y + 1$, and $z + 1$
(F) x
(G) z and x
(H) x, y, and z