2.1.

$$0, 1, 1, 2, 3, 5, 8$$

Fibonacci Numbers: $F_0 = 0$

$$F_1 = 1$$

$$F_i = F_{i-1} + F_{i-2} \text{ for } i \geq 2.$$

(a) Recursive algorithm: for k-th term.

```
int f (int k)
{
    if (n <= 1) {
        return k;
    }
    return f(n-2) + f(n-1)
}
```

(b) Dynamic programming algorithm:

→ Next page: ———————→

b)

```
int  f (int k).

  {

    int f[k + 2];      // Same extra space.

    f[0] = 0;
    f[1] = 1;

    for (int i = 0; i <= n; i++) {

      f[i] = f[i-1] + f[i-2];   // Next term
                                //   in sequence

    }

    return  f[k];

  }
```

c) ① Draw a subproblem graph

$$F(n)$$

$$F(n-1) \qquad F(n-2)$$

$$F[(n-1)-1] \quad F[(n-1)-2] \quad F[(n-2)-1] \qquad F[(n-2)-2]$$

$$\vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad\qquad \vdots$$

$$F(0) \cdots F(1) \cdots F(0) \cdots F(1) \cdots F(0) \cdots F(1)$$