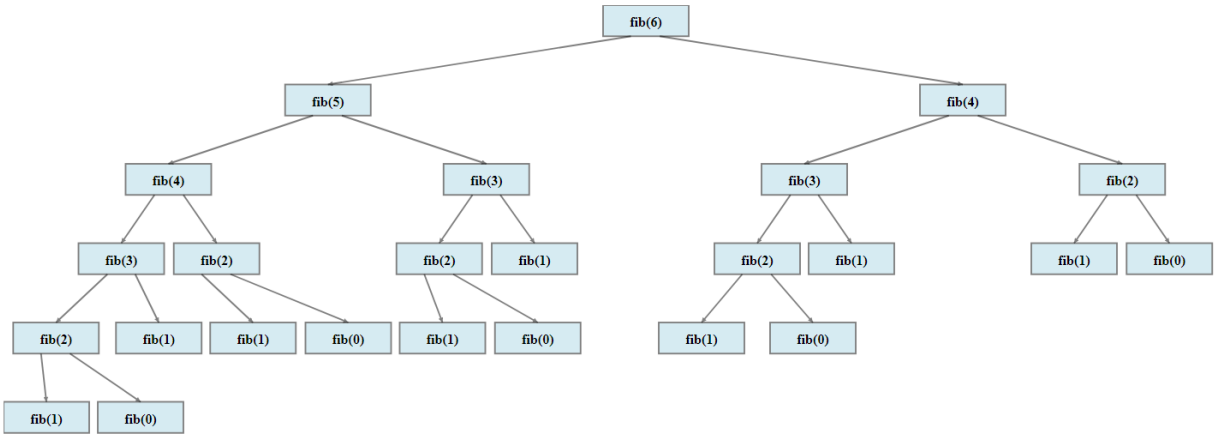
Dynamic Programming

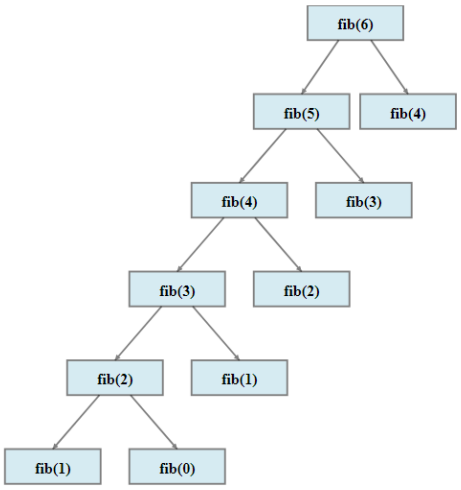
```
fib (x)
      if (x == 0) return 0
      if (x == 1) return 1
      return fib(x-1) + fib (x-2)
```

fib(6) = 8 easy
fib(60)?



Dynamic Programming to the rescue

- solve problem by breaking into smaller sub-problems
- solve each sub-problem only once
- store sub-problems solution in some DS(memoization)
- next time sub-problem occurs, use stored results



Top down approach

```
int fib(int x) {
      static int saved[1000] = {0};
// This is done for simplicity, should be DMA

      if( x==0) return 0;
      if( x==1) return 1;
      if( saved[x] != 0 ) return saved[x];
      saved[x] = fib(x-1) + fib(x-2);
      return saved[x];
}
```

0	1	2	3	4	5	6
		1	2	3	5	8

Bottom-up approach

```
int fib (int x) {
      int *saved = calloc(x+1, sizeof(int));
      int i;

      saved [0] = 0;
      saved [1] = 1;
      for (i=2;i<=x;i++) {
            saved[i] = saved[i-1] + saved[i-2];
      }
      return saved[x];
}
```

Dynamic Programming practical applications
Content awareness image resize: https://youtu.be/qadw0BRKeMk