Homework 15

2019.06.13

Note: When the exercise asks you to "design an algorithm for…," it always means that "designs an EFFICIENT algorithm for … and ANALYZES your algorithm". You should keep this in mind when writing solutions.

1. If we are willing to use a little more space (the space needed is only multiplied by a constant factor, however), it is possible to avoid the initialization time needed to set all the entries of the table to some special value. This is particularly desirable when in fact only a few values of the function are to be calculated, but we do not know in advance with ones. Show how an array $T[1..n]$ can be *virtually initialized* with the help of two auxiliary arrays $B[1..n]$ and $P[1..n]$ and a few pointers. You should write three algorithms.

```
procedure init
    { virtually initializes $T[1..n]$ }

procedure store ($i$, $v$)
    { sets $T[i]$ to the value v }

function val ($i$)
    { returns the last value given to $T[i]$ , if any;
      returns a default value (such as -1) otherwise }
```

A call on any of these procedures or functions (including a call on *init* !) should take constant time in the worst case.

2. 根據slide Ch12-1 P.19，畫出該頁TSP問題的三個版本（BFS、DFS、Best First Search）。

3. 根據slide Ch12-1 P.13，請設計一演算法，只使用一維陣列解出SOS問題(Combining DP & Backtracking)，得到一組解即可。

4. Modify the BK algorithm to branch-and-bound version to solve the maximum clique.