Problem set 8

Jonas Ishøj Nielsen, join@itu.dk

18/12/2021

Contents

1 Grid path decomposition

The implementation can be seen in `NicePathDecompostionOfGrid.py` and sets n=min(nIn,mIn) and m=max(nIn,mIn).

1. Give each node value: $v_{i,j} = i \cdot n + j$ for i=0,...,m-1 and j=0,...,n-1, s.t. it is easier to describe the order, not actually needed.
2. create leaf
3. continuously add introduce nodes until the bag $\beta(t) = \{0, ..., n\}$
4. switch between adding forget and introduce nodes. Forget the node with the lowest value and add the node next in numerical order.
5. After node $v_{m-1,n-1}$ has been added, add a series of forget nodes ending in a leaf.

This will yield a nice path decomposition with path width n.
An example can be seen in figure 1.

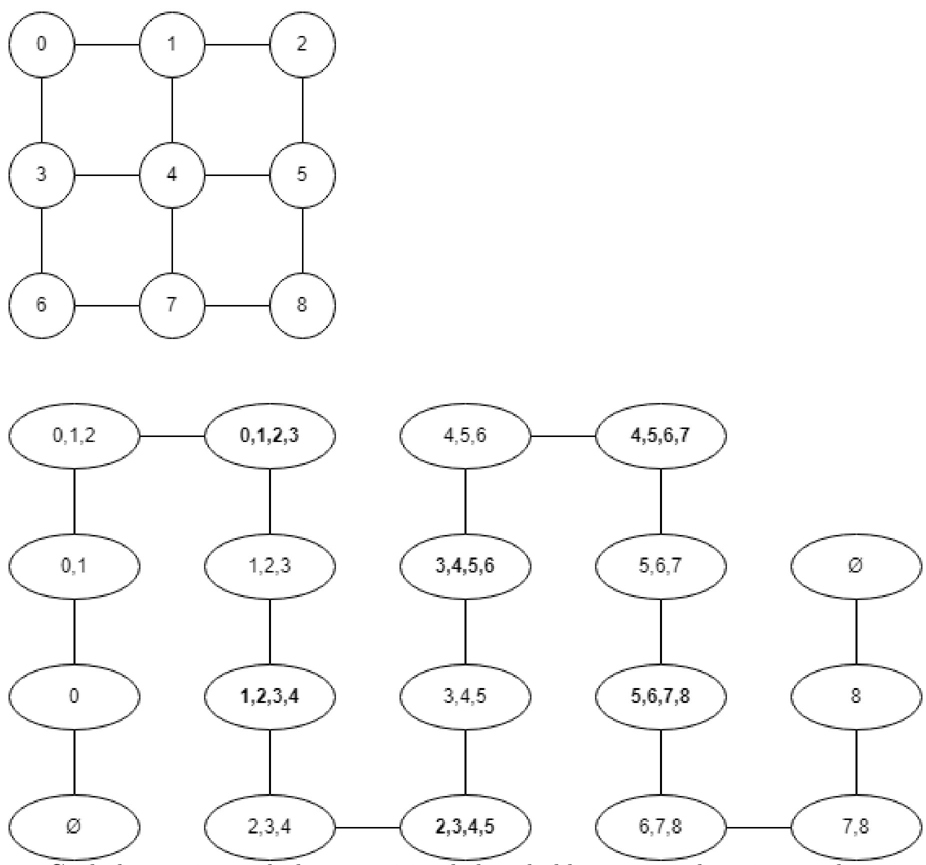

Figure 1: Grid above, nice path decomposition below, bold notes are those in tree decomposition.

2 Count #independent sets in n x n square grids

The code can be seen in `CountIndependentSetOfGrid.py` and it uses `NicePathDecompostionOfGrid.py` to get a nice path decomposition. The DP works by computing implicitly computing the table A[t,Y]=#indpendent set of G[$\gamma(t)$] for each node t and $Y \subset \beta(t)$.

- Leaf:
  A[T,Ø]=1
- Introduce v:
  $$A[T,Y] = \begin{cases} A[t,Y] & if v \notin Y \\ 0 & \text{Y is illegal} \\ A[t', Y \setminus \{x\}] & \text{else} \end{cases}$$
- Remove v:
  $A[T,Y] = A[T',Y] + A[t', Y \cup \{x\}]$

Although it works for n x m square matrices, the following results are only for n x n square matrices.
(1, 1) 2
(2, 2) 7
(3, 3) 63
(4, 4) 1234
(5, 5) 55447
(6, 6) 5598861
(7, 7) 1280128950
(8, 8) 660647962955
(9, 9) 770548397261707
(10, 10) 2030049051145980050
(11, 11) 12083401651433651945979
(12, 12) 162481813349792588536582997