

Algorithmique et programmation

Lecture 1

Dynamic programming

RNA folding

We have a $O(n^3)$ time and $O(n^2)$ space DP algorithm by considering

$$dp[i][j] = \max \left\{ \begin{array}{l} dp[i+1][j] \\ 1 + dp[i+1][k-1] + dp[k+1][j] \quad \forall k \text{ such that } i \text{ and } k \text{ form a bound} \end{array} \right\}$$

Since 2016 it is shown that it can be solved in $\tilde{O}(n^{2.8606})$ time (Bringmann et al.), and there is no $O(n^{\omega-\epsilon})$ -time algorithm unless widely believed conjecture is false, where ω is the matrix multiplication constant s.t. matrix multiplication takes $O(n^\omega)$, and we know $2 \leq \omega < 2.373$.

Knapsack

It is a *pseudo-polynomial* algorithm because it depends polynomially in n the number of things and exponentially in $\log W$ where W the weight threshold.

Questions and remarks

- Could you try not to place the camera in front of the slides because it hides the text sometimes ?