Wiktor Kuchta

4/1

Modyfikujemy algorytm z wykładu.

Inicjalizacja algorytmu jest taka sama.

Załóżmy, że mamy obliczone optymalne koszty dla poprzedniej kolumny. Wtedy tak samo jak na wykładzie możemy obliczyć optymalne koszty, jeśli z poprzedniej kolumny możemy iść w prawo albo na ukos. Zauważmy, że nigdy nie opłaca nam się zawracać, jeśli decydujemy się iść w góre lub dół. Zatem aby dla obecnej kolumny uwzględnić możliwość przechodzenia w dół, iterujemy się przez nią od góry i dla każdego pola sprawdzamy, czy koszt byłby mniejszy, gdyby dojść do niego z góry i aktualizujemy ten koszt. Analogicznie iterujemy się od dołu, aby uwzględnić możliwość przechodzenia w górę.

Optymalną trasę znajdujemy analogicznie jak na wykładzie, tylko musimy uwzględnić dodatkowe ruchy. Czyli znajdujemy pole z minimalnym kosztem w ostatniej kolumnie i szukając dla niego poprzednika na optymalnej ścieżce, sprawdzamy 5 pól, z których dało się do niego dojść.

4/3

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Korzystając z $0! = 1$, $n! \bmod p = ((n-1)! \bmod p)(n \bmod p) \bmod p$ spamiętujemy wartości silni dla argumentów niewiększych niż $m = \max n_i$.

Chcemy obliczyć odwrotności modulo p dla liczb $\{1, \dots m\}$. Jest na to wzór

$$i^{-1} = -\lfloor p/i \rfloor (p \bmod i)^{-1} \bmod p.$$

Mamy $1^{-1} \equiv 1 \pmod{p}$ i korzystając z tego wzoru obliczamy kolejno wszystkie odwrotności aż do m.

Teraz $\binom{n}{k} = (n! \bmod p)(k! \bmod p)^{-1}((n-k)! \bmod p)^{-1} \bmod p$, co można łatwo obliczyć z naszych spamiętanych liczb.

$$\begin{gathered} p \bmod i = p - \lfloor p/i \rfloor i \\ p \bmod i \equiv -\lfloor p/i \rfloor i \pmod{p} \\ i^{-1} \equiv -\lfloor p/i \rfloor (p \bmod i)^{-1} \pmod{p} \end{gathered}$$