Pockets
Rozwiązanie Wzorcowe

Marek Żylak

1 Wstęp

Rozwiązanie wzorcowe `h.cpp` składa się z dwóch faz:

1. symulacji składania kartki
2. liczenia kieszonek

2 Symulacja składania kartki

Na początku zauważmy, że zagięcie lewej części figury w prawo można zastąpić zagięciem prawej części figury w lewo. Trzeba tylko zapamiętać, że po takiej operacji blat stołu na którym leży figura przylega do jej przeciwnej strony.

Do reprezentacji aktualnego stanu składanej kartki używana jest tablica odpowiadająca podziałowi stołu na pola. Na każdym polu może leżeć stos jednostkowych kawałków kartki, reprezentowanych jako lista dwukierunkowa. Dodatkowa dla każdego kwadratu jednostkowego pamiętana jest jego orientacja (może być odbity niezależnie poziomie lub pionie). Dokonanie złożenia polega po prostu na przekładaniu kwadratów jednostkowych z jednych pól na inne.

Ilość złożeń jest $O(N)$, a w każdym złożeniu trzeba przełożyć $O(N^2)$ kwadratów jednostkowych zatem złożoność algorytmu wynosi $O(N^3)$.

Łatwo widać, że da się wykonywać takie złożenia, żeby figura wychodziła poza pola na, które początkowo zajmuje na stole. Co więcej można oddalić się od początkowego położenia o $\Theta(N^2)$ jednostek, zatem niepraktyczne jest rozwiązanie polegające na alokowaniu tak dużej tablicy pól, żeby figura z niej nie uciekła — taka tablica musiała by mieć $\Theta(N^4)$ elementów. Istnieją co najmniej dwa wyjścia z tej sytuacji:

- Można skorzystać ze spostrzeżenia wspomnianego na początku rozdziału żeby, wykonywać zawsze tylko takie operacje, które zwalniają pola w tablicy. W tym przypadku trzeba jednak dodatkowo pamiętać po której stronie kartki jest stół i jaką ma orientację. Dodatkowo, z powodu poruszania się stołu, powstają sztucznie operacje składania kartki pod spód, gdy w treści zadania zawsze jest mowa o składaniu na wierzch. Mimo wszystkich trudności, da się opisany pomysł elegancko zakodować, unikając rozpatrywania symetrycznych przypadków. Trzeba tylko intensywnie stosować parametryzację i tablicowanie częściowych wyników (np. zamiast w elemencie listy trzymać wskaźniki `next` i `prev`, lepiej zrobić tablicę `link[2]`). Taką metodę zastosowano w rozwiązaniu wzorcowym.

- Wynik złożenia można zapisywać po prostu do pomocnieczej tablicy pól. Autor rozwiązania wzorcowego jest przekonany, że łatwiej da się napisać bezłędny program implementujący tą metodę. Nie została on zastosowania przez autora tylko dlatego, że wcześniej wymyślił i zakodował pierdzą metodę.

3 Liczenie kieszonek

Aby wyznaczyć ilość kieszonek w danej krawędzi trzeba znaleźć strukturę zachodzenia na siebie papieru w tym miejscu. Na krawędzi kwadratu jednostkowego możemy spotkać się z dwoma sytuacjami:

- dwa kwadratu jednostkowe, połączone krawędzią zakrywają wszystko pomiędzy nimi
- krawędź kwadratu jednostkowego, która jest zarazem brzegiem kartki

Łatwo widać, że dwa kwadraty jednostkowe połączone krawędzią kryją wszystkie kieszonki wytworzone przez kwadraty znajdujące się między nimi. Można szybko dojść do wniosku, że strukturę tego zakrywania można reprezentować przy pomocy nawiasów. Odpowiednią sekwencję nawiasów budujemy przechodzą przez wszystkie kwadraty jednostkowe w kolejności ich ułożenia na ostatnim polu. Gdy trafiamy na pierwszy kwadrat, z pary sąsiadów połączonych krawędzią, wstawiamy nawias otwierający, a gdy spotykamy drugi kwadrat z tej pary wstawiamy nawias zamykający. Krawędź brzegową można opisać przy pomocy pary następujących po sobie nawiasów — otwierającego i zamykającego. W tak utworzonym napisie liczymy w ilu miejscach łączą się ze sobą wyrażenia nie będące już częścią. Te miejsca odpowiadają kieszonkom, bo odpowiadają im szczeliny między dwoma kolejnymi ścianami, do których dojście nie jest zablokowane przez papier.

4 Podsumowanie

Całkowita złożoność rozwiązania wzorcowego wynosi $O(N^3)$.