Rozwiązanie wzorcowe zadania F/2006

Autor: Tomasz Kazana

OPRACOWANIE ZADANIA

Rozwiązanie polega na zbadaniu pewnej liczby przypadków, wśród których na pewno znajduje się rozwiązanie optymalne.

Z treści zadania wiemy, że rozwiązanie optymalne składa się z dwóch ciągów o pewnym wspólnym prefiksie, tzn. wiemy, że rozwiązanie to ciągi AX i AY, gdzie A, X i Y są pewnymi rozłącznymi ciągami o szczególnych własnościach - element ciągu to pewna zębatka bądź para zębatek. W pierwszej fazie algorytmu generowane są wszystkie możliwe ciągi liczb 1 i 2 o sumie N (przypomnijmy: N to liczba zębatek). Teraz: dla ustalonego wzorca (przez wzorzec rozumiemy tu rozbicie liczby N na składniki 1 i 2, np. $6 = 2 + 2 + 1 + 1 = 1 + 2 + 1 + 2$), zakładamy że jest on postaci A'X'Y'Z' (Z oznacza nie wykorzystane zębatki), gdzie A',X',Y' i Z' stanowią wzorce odpowiadających im ciągów A,X,Y i Z. Podobnie: dla ustalonej permutacji zębatek π przyjmujemy, że konkatenacja ciągów A, X, Y i Z daje π. Ponieważ analizujemy wszystkie możliwe rozbicia wzorca na czwórki i wszystkie możliwe permutacje zębatek, więc optymalne rozwiązanie musi być rozważone.

Dla każdego badanego przypadku najpierw sprawdzamy, czy rzeczywiście ciąg AX generuje obroty odpowiednie dla wskazówki minutowej, a AY - dla godzinowej. Jeśli nie, rozważamy kolejny przypadek. Jeśli tak - porównujemy (wg podanych kryteriów) z dotychczas najlepszym wynikiem i ewentualnie zapamiętujemy najlepsze dotychczasowe rozwiązanie.

Na koniec programy wypisują wynik według podanego schematu. W szczególności w rozwiązaniu optymalnym na ostatniej wspólnej osi zębatek mogą być trzy zębatki - nie ma sprzeczności z opisaną wcześniej procedurą, ponieważ zakładamy, że ostatni element A, pierwszy element X, a także pierwszy element Y, mogą znajdować się na wspólnej osi.

ANALIZA ZŁOŻONOŚCI

Ograniczenia w zadaniu są bardzo małe ($N \leqslant 6$), więc ilość rozważanych przypadków jest mała, pomimo generowania wszystkich możliwych permutacji i wszystkich rozbić wszystkich wzorców. Zgrubnie szacując: co najwyżej 6! permutacji, co najwyżej 13 wzorców, co najwyżej $\binom{6}{3} = 20$ rozbić ustalonego wzorca, co daje co najwyżej 187200 przypadków. Każdy przypadek jest sprawdzany liniowo ze względu na N (wymaga przeliczenia obrotów kolejnych osi wg podanego wzoru).

UWAGI IMPLEMENTACYJNE

W implementacji korzystamy z dwóch różnych funkcji sprawdzających oddzielnie:
gdy $Y = \epsilon$ używana jest funkcja `probuj(vector<string>)`, a w pozostałych przypadkach: funkcja `probuj(vector<string>, vector<string>, int)`.

Do generowania wszystkich permutacji, w kodzie C++ korzystamy z funkcji bibliotecznej next_permutation, a w kodzie Java - z autorskiej implementacji tej funkcji.