Algorytm sortowania butelek z atramentem przez maszynę

Hubert Fedorowicz

Uruchamianie:

* Windows - program.exe;
* Linux - make all i solveInkProblem.

Treść zadania:

Na półce są ułożone w przypadkowej kolejności pojemniki z atramentami w czterech kolorach: C, M, Y i K. Należy je uporządkować za pomocą robota w taki sposób, aby od lewej strony półki znalazły się kolejno: C, następnie M, następnie Y, a na końcu - K. Robot w jednym ruchu może pobrać jednocześnie dokładnie cztery sąsiednie pojemniki z dowolnego miejsca na półce i przenieść je na prawą stronę, za wszystkie pojemniki, a następnie dosunąć część pojemników w lewo, wypełniając lukę. Ułożyć plan pracy robota, dyktujący w kolejnych ruchach, którą czwórkę pojemników przenieść na prawą stronę. Wykonać jak najmniejszą liczbę ruchów. Porównać czas obliczeń i wyniki różnych metod.

Rozwiązanie problemu:

Ze względu na to, że przy każdym przeniesieniem czwórki butelek zmieniane jest miejsce wszystkich butelek na prawo od miejsca pobrania butelek rozsądnym rozwiązaniem było by sortowanie butelek od lewej strony. Algorytm mógłbym sortować po kolei butelki poczynając od pierwszej od lewej i zmniejszać problem o jeden element do sortowania. Problem pojawia się, gdy do posortowania zostaną nam ostatnie cztery butelki. Problem sortowania mniej niż czterech butelek za pomocą opisanej w zadaniu maszyny nie jest możliwy, ponieważ nie mamy możliwości zmienić ich położenia. Przy pięciu butelkach maszyna ma możliwość jedynie przesuwania czterech pierwszych butelek od lewej na prawą stronę, a piątą na pierwsze miejsce od lewej, tak więc przy pięciu butelkach mamy możliwość posortowania tylko pewnego zbioru przypadków. Dopiero od sześciu butelek mamy możliwość posortowania każdego możliwego ułożenia butelek. Z tego powodu podzieliłem problem na dwie części:

* sortowanie ostatnich sześciu butelek od prawej.
* sortowanie pozostałych butelek.

W pierwszej kolejności sortujemy butelki od lewej strony (bez ostatnich sześciu). Jest to stosunkowo proste. Należy to zrobić w następujących krokach:

* Sprawdzić jaki powinien być typ atramentu w sprawdzanej butelce.
* Wyszukać najbliższej butelki z szukanym atramentem znajdującej się w odległości od aktualnej pozycji będącej wielokrotnością liczby 4 (licząc łącznie z aktualnym polem) - taką butelkę można najłatwiej przemieścić.
* Jeśli jest taka butelka przenieść wszystkie czwórki pomiędzy tą butelką, a miejscem docelowym. Następnie rozwiązać problem dla butelek na prawo od przestawionej butelki.
* jeśli nie ma takiej butelki:
  + Sprawdzić które miejsce z ostatnich czterech od prawej znajduje się w odległości od docelowego miejsca, będącej wielokrotnością liczby cztery.
  + Umieścić butelkę z pasującym atramentem w znalezione miejsce.
  + Przenieść czwórki butelek znajdujące się pomiędzy znalezioną butelką a miejscem docelowym.
  + Rozwiązać problem dla butelek na prawo od przestawionej butelki.

Rozwiązanie problem sortowania ostatnich sześciu butelek jest trudniejsze, ponieważ każde przemieszczenie butelek istotnie zmienia stan ułożenia butelek. Problem rozwiązałem przez wygenerowanie rozwiązań, dla wszystkich nieposegregowanych ustawień sześciu butelek. Zrobiłem to przez wygenerowanie wszystkich możliwych wyników jakie można uzyskać przestawiając za pomocą zadanej maszyny sześć butelek ponumerowanych od lewej ('1','2','3','4','5','6'). Generacja polegała na stworzeniu zbioru rozwiązań, początkowo składającego się z wspomnianej wcześniej półki z ponumerowanymi kolejno butelkami i dodawaniu do zbioru półek uzyskanych przez wykonanie na półkach ze zbioru wszystkich możliwych do wykonania operacji, które reprezentują rozwiązanie, które jeszcze nie znajduje się w zbiorze, lub znajduje, ale jest uzyskane przez mniejszą liczbę ruchów niż to ze zbioru (w takim wypadku rozwiązanie jest zastępowane). Taki zbiór na końcu składał się z 360 elementów, co jest połową wszystkich możliwych ułożeń - permutacja 6'elementowa bez powtórzeń zbioru [1,2,3,4,5,6} = 720.

Następnie wygenerowałem zbiór możliwych nieposortowanych półek z sześcioma butelkami - permutacja 6'elementowa z powtórzeniami zbioru {'C','M,'Y','K'} bez tych już posortowanych. Dla każdego elementu zbioru znalazłem zbiór możliwych rozwiązań a następnie z wcześniej wygenerowanego zbioru możliwych wyników wybierałem ten najlepszy i przypisywałem listę przesunięć do danej sytuacji i dodawałem ją do listy sytuacji i rozwiązań. W ten sposób uzyskałem listę wszystkich nieposegregowanych sytuacji i ruchów potrzebnych do jej rozwiązania - 6result.txt.