Zadanie: skreślanie ciągów

Przypuśćmy, że mamy dany ciąg znaków, który chcemy wytrzeć. Nie jest to jednak tak proste, jak mogłoby się wydawać, bo (z powodu klątwy faraona) jesteśmy ograniczeni – możemy wycierać tylko niektóre znaki oraz niektóre wzorce składające się z dwóch znaków.

Przykłady: Dla ciągu xAAxB oraz wzorców x, xA, xB, AB:
wycieramy xA, zostaje nam AxB,
wycieramy x, zostaje AB,
wycieramy AB i osiągamy sukces.
(Zauważmy, że nieprzemyślane wytarcie xB na początku spowoduje, że sukces nie będzie możliwy).

Dla ciągu ())())(() oraz wzorca () nie da się osiągnać sukcesu.
Po trzykrotnym wytarciu wzorca zostaje nam ciąg))(i jest to najkrótszy ciąg, jaki możemy uzyskać.
Zadanie składa się z trzech części:

1. Dla zadanego ciągu znaków oraz listy wzorców określ, czy możliwe jest wytarcie wszystkich znaków przez wycieranie podanych wzorców (wartość zwracana przez metodę Erasable).
2. Wyznacz najmniejszą możliwą liczbę wytarć potrzenych do wytarcia wszystkich znaków. Jeżeli nie jest to możliwe, zwróć int.MaxValue (parametr wyjściowy crossoutsNumber metody Erasable).
3. Wyznacz długość najkrótszego ciągu, jaki można uzyskać przez wycieranie podanych wzorców (wartość zwracana przez metodę MinimumRemainder).

Punktacja:
Etap 1. 1 punkt,
Etap 2. 0.5 punktu,
Etap 3. 1 punkt.
Wskazówki:
Zastosuj programowanie dynamiczne.
Dla każdego spójnego fragmentu podanego ciągu oblicz, czy fragment ten może być wytarty (skreślony).
Fragmenty indentyfikuj poprzez indeksy początku i końca.
Fragment tekstu jest wycieralny, jeżeli:

- jest pusty
- jest jednym ze wzorców
- jest konkatenacją dwóch krótszych fragmentów wycieralnych
- zawiera (spójny) fragment wycieralny, a to co zostaje po jego usunieciu jest jednym ze wzorców

Uwagi:

- Zastanowić się jak inaczej można opisać ostatnią wymienioną powyżej cechę tekstów wycieralnych – to znaczy, jakie powinno być wzajemne położenie wycieralnego fragmentu i pozostałego wzorca (lub jego części). Niektóre potencjalne położenia zawierają się już we wcześniejszych własnościach (i nie trzeba ich ponownie sprawdzać). Jeśli do implementacji sprawdzenia ostatniej cechy potrzebna Ci jest pętla, to znaczy że nie wybrałeś najprostszego sposobu.
- Wygodnie jest skorzystać z podanych funkcji pomocniczych comparePattern.