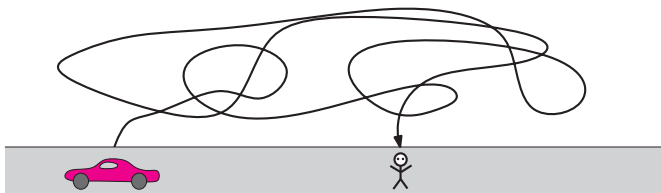


Krowa, las i eksploracja terenu

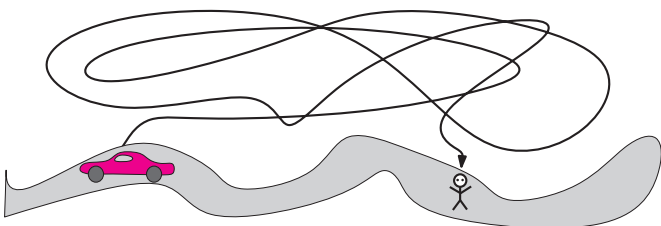
Marcin BIEŃKOWSKI Instytut Informatyki
Uniwersytetu Wrocławskiego

Ten artykuł został zapożyczony z Wrocławskiego Portalu Informatycznego <http://informatyka.wroc.pl>. Więcej informacji na temat portalu można znaleźć na tylnej okładce numeru.

Kilkanaście lat temu zgubiłem się w lesie. Nie na tyle, żeby sytuacja była beznadziejna: wiedziałem, że zostawiłem auto przy drodze, wszedłem do lasu, gdzie spędziłem jakiś czas, chodząc, a następnie wróciłem do tej samej drogi. Dodatkowo, podczas spaceru po lesie przeciąłem swoją drogę dostatecznie dużo razy, żeby nie mieć zielonego pojęcia, czy samochodu należy szukać, idąc w lewą stronę czy w prawą. Ot, mniej więcej tak jak na poniższym rysunku.



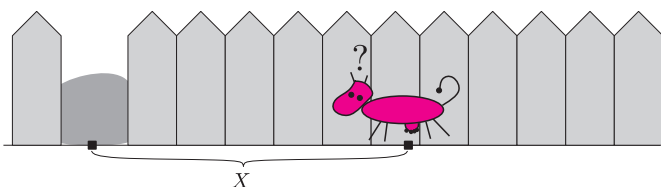
No dobrze, to jest zły rysunek. Tak naprawdę sytuacja wyglądała znacznie gorzej, a mianowicie tak:



To znaczy, drzewa na tyle skutecznie zasłaniały drogę, że widoczność była praktycznie zerowa. Nie miałem szans zobaczyć samochodu z odległości, musiałem się na niego natknąć.

Parę lat później (pomińmy chwilowo milczeniem to, ile się nachodziłem w drodze do samochodu) dowiedziałem się, że ktoś już wcześniej miał taki sam problem i zostało to dokładnie opisane w literaturze. A mianowicie, ten sam problem miała pewna krowa.

Krowa jaka jest, każdy widzi. Krowa stoi przy prostym płocie, w którym jest wejście na pastwisko. Wejście (które jest oddalone od krowy o co najmniej jeden metr) krowa zobaczy dopiero, gdy do niego dotrze. Co powinna zrobić, żeby jak najszybciej rozpocząć zjadanie trawy? Problem ten jest jednym z najprostszych wariantów zagadnienia eksploracji terenu. Jak okaże się poniżej, prostym, lecz nie trywialnym.



Postaramy się wyrobić sobie podstawowe intuicje. W całym artykule przez X będziemy oznaczać

początkową odległość, w metrach, dzielącą krowę od wejścia na pastwisko. Oczywiście, absolutnie genialna krowa, znająca mapę terenu, potrafi dojść na pastwisko, przeszedłszy X metrów. Co się dzieje, jeśli krowa jest trochę głupsza, czyli nie zna mapy, nie zna X , ale wie, że wejście na pastwisko jest z prawej strony? Oczywiście, idąc uparcie w tym kierunku, przejdzie też dokładnie X metrów.

A co ma zrobić krowa, jeśli w jakiś sposób poznała wartość X (przykładowo, wie, że od wymarzonego celu dzieli ją 10 metrów), ale nie wie, czy wejście jest z lewej, czy z prawej? Może wtedy zastosować następujący algorytm z parametrem $y = X$.

Algorytm Go-and-Eat(y)

1. Idź y metrów w lewo
2. Jeśli nie znajdziesz wejścia, zawróć do punktu startowego
3. Idź y metrów w prawo

Łatwo obliczyć, że w najgorszym przypadku (a tylko takie przypadki interesują biedne, pechowe krowy) krowa przejdzie $3X$ metrów (czyli w naszym przykładzie trzydzieści). Uważny Czytelnik zauważy, że to, czy krowa idzie na początku w lewo czy w prawo, w pewnym sensie nie ma znaczenia: jeśli krowa ma pecha, to wejście będzie i tak z drugiej strony niż ta, którą wybierze.

Krowa głodna to krowa zła. Prawdziwy problem krowy (który jest identyczny z moim poszukiwaniem samochodu) zaczyna się jednak wtedy, kiedy krowa nie zna ani wartości X , ani nie wie, z której strony jest wejście. Zamiast wyciągnąć gotowe rozwiązanie z kapelusza, zastanówmy się, co mogłoby pomóc krowie w tym zadaniu.

A gdyby krowa знаła jakieś oszacowanie na X , na przykład znała pewną liczbę $D \geq X$? Mogłaby uruchomić procedurę Go-and-Eat(D), tj. iść D metrów w lewo, wrócić do punktu wyjścia, a następnie iść w prawo. Przebyta droga w najgorszym przypadku wynosiłaby $2D + X$. Czyli jeśli D byłoby „tylko trochę” większe od X , to otrzymalibyśmy rozwiązanie niedużo gorsze niż w przypadku, kiedy znamy X . No dobrze, ale jak znaleźć takie D ? Pomoże nam w tym często stosowana w algorytmice **metoda podwajania**: uruchamiamy algorytm Go-and-Eat kolejno z wartościami 1, 2, 4, 8, 16, ... Cieszą nas dwie własności.

- Jeśli w końcu uruchomimy Go-and-Eat z potęgą liczby 2 większą od X , to krowa znajdzie pastwisko.
- Sumaryczna droga przebyta podczas uruchamiania Go-and-Eat ze zbyt małymi wartościami parametru jest niezbyt długa. (To dopiero wykażemy.)

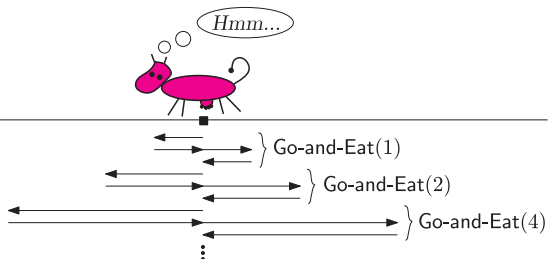
Ale moment! – powinien krzyknąć Uważny Czytelnik. Przecież po uruchomieniu algorytmu Go-and-Eat(1) krowa znajdzie się o 1 metr od punktu wyjścia, a algorytm Go-and-Eat(2) ma szansę zadziałać pod warunkiem, że krowa znajduje się początkowo w punkcie startowym. Możemy „załatać” algorytm, dodając do niego dodatkowy krok:

4. Zawróć do punktu startowego

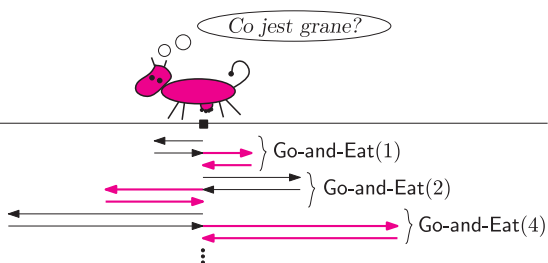
Łata na łacie. Tu mała dygresja – informatycy, jak i wszystkie umysły ściśle, uwielbiają: a) łątanie złych rozwiązań zamiast pisanie ich od nowa, b) redukcje. Do tego stopnia uwielbiają, że często wpadają w pułapki jak w opisanej niżej historii (z długą brodą) o czajniku:

Mamy pusty czajnik, chcemy zagotować wodę. Co zrobić? Należy oczywiście: 1. nalać wody, 2. włączyć czajnik, 3. czekać. A co zrobić, jeśli mamy pełny czajnik i chcemy zagotować wodę? To proste – odpowie prawdziwy (i leniwy) informatyk – wylewamy z niego wodę i w ten sposób doprowadzamy sytuację do przypadku, który potrafimy rozwiązać.

Zastanówmy się zatem, czy przez „łątanie” algorytmu Go-and-Eat i dodanie punktu 4. nie postąpiliśmy przypadkiem jak prawdziwi informatycy (a nie chcielibyśmy tego robić, bo po co męczyć steraną życiem i mocno już wygłodniałą krowę). Zobaczmy, jak wygląda droga, po której teraz chodzi krowa.



Może nie jest oczywiste, co można tutaj poprawić. Ale... pamiętajcie, że to, w którą stronę krowa zaczyna wykonywanie algorytmu Go-and-Eat, nie ma znaczenia? W takim razie narysujmy drogę krowy, która nieparzyste wykonania algorytmu Go-and-Eat zaczyna w stronę lewą, a parzyste w prawą.

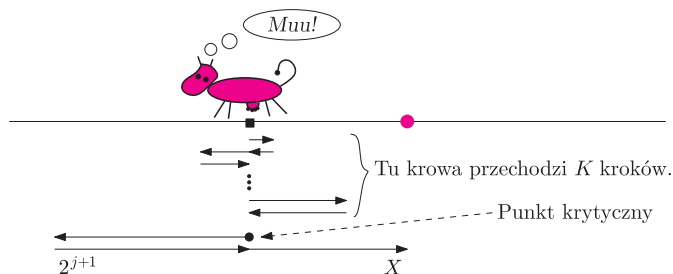


Bez sensu, prawda? Krowa idzie w jedną stronę 2^j metrów tylko po to, żeby wrócić do punktu wyjścia i pójść w tę samą stronę 2^{j+1} metrów! Wyrzucmy zatem nadmiarowe spaceru (zaznaczone

kolorem na poprzednim rysunku). Zmodyfikowany algorytm prezentuje się następująco:

Algorytm Smart-Cow

1. Kierunek zwiedzania := lewo
2. $D := 1$
3. Dopóki nie znalazłeś wejścia na pastwisko, powtarzaj:
 - a. Przejdź D metrów w kierunku zwiedzania i wróć do punktu startowego
 - b. $D := D \cdot 2$
 - c. Odwróć kierunek zwiedzania



Spróbujmy teraz przeanalizować, jaki dystans pokona krowa w najgorszym przypadku. Weźmy takie naturalne j , że zachodzi $2^j < X \leq 2^{j+1}$. Oznacza to, że pierwsze j wykonania pętli w punkcie 3. powyższego algorytmu (z wartościami $D \leq 2^j$) nie spowoduje znalezienia pastwiska. Droga przebyta przez krowę do tego punktu wynosi

$$\begin{aligned} K &= (1 + 1) + (2 + 2) + (4 + 4) + (8 + 8) + \\ &\quad + \dots + (2^{j-1} + 2^{j-1}) + (2^j + 2^j) = \\ &= 2 \cdot (1 + 2 + 4 + 8 + \dots + 2^{j-1} + 2^j) = \\ &= 2 \cdot (2^{j+1} - 1) < 4 \cdot 2^j. \end{aligned}$$

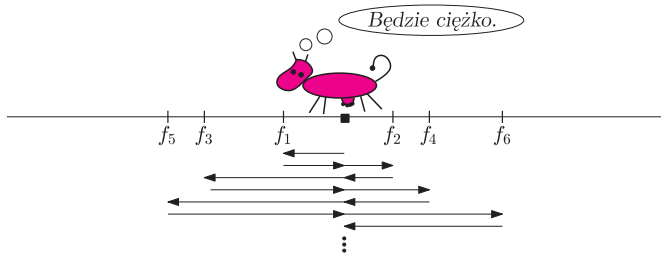
Nazwijmy ten punkt *krytycznym*, przyda się on w dalszej części artykułu. Następnie krowa chce iść 2^{j+1} metrów w jakimś kierunku. Oczywiście, ma pecha i idzie w kierunku przeciwnym do wejścia na pastwisko. Ten skok w niewłaściwy bok (wraz z powrotem) kosztuje ją kolejne $2^{j+1} + 2^{j+1} = 4 \cdot 2^j$ metrów. Następnie krowa zamierza iść 2^{j+2} metrów, ale już po X metrach napotyka pastwisko. Jej całkowita droga to co najwyżej

$$K + 4 \cdot 2^j + X < 8 \cdot 2^j + X < 8 \cdot X + X = 9X.$$

Czy naprawdę trzeba tyle chodzić? Krótka odpowiedź brzmi: tak, każdy algorytm dla krowy wymaga w najgorszym przypadku przejścia odległości dziewięciokrotnie przekraczającej początkową odległość krowy od pastwiska. Niestety, wykazanie tego jest dość trudne i wykracza poza ramy tego artykułu. Spróbujmy jednak uspokoić swoje sumienie i sprawdzić, że krowa zawsze musi przejść co najmniej dystans $5X$.

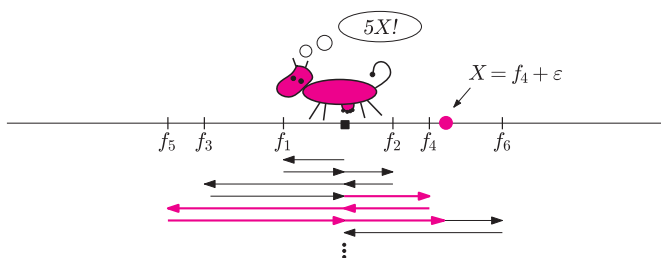
W tym celu zauważmy, że jeśli krowa nie robi głupich wycieczek (takich jak na dwóch wcześniejszych rysunkach), to wędruje, oddalając się od punktu startowego na przemian w lewo i w prawo.

Wyznacza to podział trasy krowy na kolejne fazy. Bez straty ogólności założmy, że na początku krowa idzie w lewo. Wtedy jej trasę można opisać liczbami $f_1, f_2, f_3, f_4, \dots$, gdzie f_j jest oddaleniem w fazie j . (W opisanym wcześniej algorytmie Smart-Cow mamy $f_j = 2^{j-1}$.)



Zauważmy teraz, że krowa wybiera taką samą trasę niezależnie od tego, jakie jest X . Powyższe zdanie wydaje się oczywiste (jak trasa mogłaby od tego zależeć, skoro X poznajemy dopiero w momencie odnalezienia pastwiska, a wtedy już nigdzie nie chodzimy?), ale warto je przeczytać więcej niż raz. Oznacza to, że dla danego wyboru trasy krowy (tj. liczb $f_1, f_2, f_3, f_4, \dots$) wystarczy postawić bramkę na pastwisko w jakimś bardzo złośliwym miejscu.

Przypatrmy się bliżej ciągowi $f_1, f_2, f_3, f_4, \dots$. Musi w nim istnieć jakaś taka kolejna para f_i i f_{i+1} , że $f_i \leq f_{i+1}$. Na powyższym rysunku jest to, na przykład, para f_4 i f_5 . W przeciwnym przypadku ciąg f_i byłby ściśle malejący (a jest to dość kiepski pomysł, jeśli krowa zamierza znaleźć odległe pastwisko). Założmy, że odległość f_i krowa przechodzi w prawą stronę. Postawmy zatem bramkę kawałek dalej niż w odległości f_i po prawej stronie od punktu startowego, tj. $X = f_i + \varepsilon$.



Wtedy krowa przechodzi co najmniej $2 \cdot f_i$ w prawo, potem $2 \cdot f_{i+1}$ w lewo, a następnie (pod warunkiem, że nie jest bezdennie głupia i $f_{i+2} > f_i + \varepsilon$) krowa dochodzi do pastwiska, przeszedłszy odległość X . Całkowita droga przebyta przez krowę to

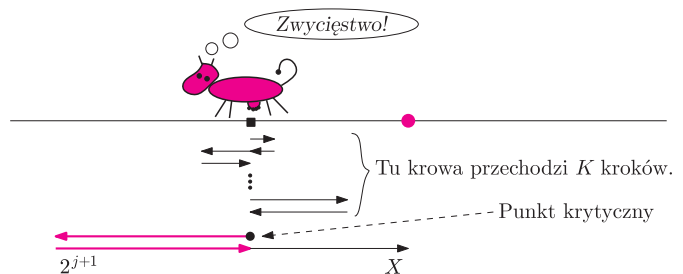
$$2 \cdot f_i + 2 \cdot f_{i+1} + X \geq 4 \cdot f_i + X = 5X - 4\varepsilon.$$

Ponieważ ε możemy wybrać dowolnie małe i zaniedbywalne, krowa przechodzi praktycznie pięciokrotnie więcej niż X .

Po co krowie pieniądze? Jak można ułatwić życie naszej krowie? Na początku poprzedniego rozdziału powiedzieliśmy, że nie da się poprawić wyniku osiąganego przez algorytm Smart-Cow. Cóż, jest to

tylko częściowa prawda. Nie można go poprawić, jeśli krowa chodzi *deterministycznie*. Możemy natomiast podarować jej monetę, którą może wspomóc się w trudnych chwilach (i nie chodzi nam o to, żeby kupiła sobie za to mapę czy coś do jedzenia).

Wróćmy do algorytmu Smart-Cow. Założmy, że w momencie, który nazwalibyśmy krytycznym, krowa rzuca monetą. Jeśli wypadnie orzeł, idzie następnie 2^{j+1} metrów w prawo, w przeciwnym przypadku w lewo. Dla ustalonej pozycji bramki na pastwisko krowa ma $1/2$ szansy na to, że pójdzie w jej stronę. W takim przypadku części spaceru krowy (kolorowy fragment rysunku) o długości $2 \cdot 2^{j+1}$ w ogóle nie ma! Wtedy krowa przechodzi nie $9X$, a $5X$. W średnim przypadku przechodzi zatem $7X$.



To obniża średnią liczbę metrów przebytą przez krowę, pod warunkiem że... krowa wymyśli, iż dany moment jest krytyczny! Okazuje się jednak, że wcale nie musi tego robić. Wystarczy, żeby rzuciła monetą, wybierając kierunek na samym początku. Wtedy analiza jest dokładnie taka sama, a dodatkowo, w momencie kiedy krowa znajdzie się już w punkcie krytycznym, z prawdopodobieństwem $1/2$ będzie szła w lewo, a z prawdopodobieństwem $1/2$ – w prawo.

Quo vadis, droga krowo? Powyższe rozważania to tylko czubek góry lodowej tematyki eksploracji terenu. Oprócz „prawdziwych” zastosowań, istnieje jeszcze parę lepiej lub gorzej zbadanych podstawowych problemów. Przykładowo, Czytelnik Dociekliwy może zastanowić się, jaką strategię należy obrać, jeśli jest się kapitanem, który odpłynął od brzegu i stracił orientację. Zakładamy, że linia brzegowa jest linią prostą i chcemy dopłynąć do jej dowolnego miejsca. Możemy wyobrazić sobie następujące warianty tego problemu:

- Wiadomo, że linia brzegowa jest oddalona o 1 km, ale nie wiadomo, w którym kierunku.
- Mamy kompas i wiemy, że linia brzegowa przebiega albo z północy na południe, albo z zachodu na wschód, nie znamy natomiast odległości od linii brzegowej. (Warto zauważyć, że jeśli wiedzielibyśmy np., że linia brzegowa przebiega z północy na południe, to jest to taki sam problem, jaki miała nasza krowa.)
- Nic nie wiemy.

Temperować ołówki, siodłać krowy i do dzieła!