Kombinatoryczna teoria liczb

Teoretyczny opis problemu

Anna Zawadzka
Piotr Waszkiewicz
Przemysław Rząd

18 listopada 2016

Cel projektu

Projekt ma na celu zaimplementowanie gry w **rozdzielanie Szemerediego**, w której gracz mierzy się z komputerem. Dla wybranych na początku gry wartości liczb **N** i **k**, runda polega na:

- Wyborze dwóch (dotąd niewybranych) liczb ze zbioru [N] przez gracza pierwszego
- Wyborze jednej ze wskazanych liczb przez gracza drugiego, która zostanie pokolorowana na jego kolor. Druga ze wskazanych liczb kolorowana jest na kolor gracza pierwszego

W rundach nieparzystych role się zamieniają. Wygrywa gracz, który pierwszy będzie miał k-elementowy ciąg arytmetyczny w swoim kolorze.

Twierdzenie Szemerediego

W projekcie będziemy korzystać z twierdzenia Endre Szemerédiego, które brzmi następująco:

Dla dowolnej liczby $0<d<1$ zwanej gęstością i dowolnej liczby naturalnej k istnieje liczba $N(d,k)$ taka, że jeżeli $N>N(d,k)$, to dowolny podzbiór A zbioru $\{1,...,N\}$ o liczebności większej od dN zawiera ciąg arytmetyczny długości k.

Inne sformułowanie twierdzenia operuje liczbą największego podzbioru zbioru [N] niezawierającego żadnego ciągu arytmetycznego o długości k. Z górnego oszacowania tej liczby skorzystamy przy sprawdzaniu, jak długo może toczyć się gra bez zwycięstwa.

Sprawdzenie, czy gra skończy się wygraną

Twierdzenie Szemerediego zostanie wykorzystane do sprawdzenia, czy na pewno gra zostanie skończona czyjąś wygraną, na podstawie podanych wartości liczb **N** i **k**. W przypadku, gdy nie ma gwarancji na zakończenie gry, odpowiedni komunikat zostanie wyświetlony użytkownikowi.

Dla danych wartości **N** i **k** gra zawsze skończy się czyjąś wygraną, jeśli najliczniejszy podzbiór niezawierający żadnego ciągu arytmetycznego o długości **k** ma liczność mniejszą niż **N**/2.

Innymi słowy, dowolny podzbiór o liczności **N**/2 zawiera jakiś ciąg arytmetyczny długości **k**.

Czyli, korzystając z górnego oszacowania **N(k, d)**, stwierdzimy że gra zawsze skończy się wygraną, jeśli spełnione będzie:

$$N > 2^{2^{d^{-2^{2^{k+9}}}}}$$

Gdzie:

$$d = \frac{1}{2} - \frac{1}{N}$$

(Co odpowiada przedostatniej możliwej rundzie, po której pozostaną tylko dwie niewybrane liczby)

Poniższa tabela przedstawia wymagane minimalne wartości **N**, aby dla podanej wartości liczby **k** mieć pewność, że gra zakończy się czyjąś wygraną:

k	**N**
3	9
4	35
5	178
6	1132

Sprawdzenie, jak długo może się toczyć gra

Zakładając, że gracze kooperują ze sobą w celu jak najdłuższego braku osiągnięcia wygranej, to znaczy starają się wybierać liczby tak, aby odwlekać czyjąkolwiek wygraną, spróbujemy oszacować jak długo (ile rund) może toczyć się taka rozgrywka.

Dla danych **N** i **k** problem ten polega na znalezieniu liczności **R** najliczniejszego podzbioru zbioru [**N**] niezawierającego żadnego ciągu arytmetycznego o długości **k**.

Korzystając z górnego oszacowania przyjmujemy, że gra może się toczyć co najwyżej **R** rund, gdzie:

$$R \leq \frac{N}{(\log \log N)^{2^{-2^{k+9}}}}$$

Wyszukiwanie najdłuższego ciągu arytmetycznego

Znalezenie najdłuższego ciągu arytmetycznego spośród liczb wybranych przez gracza (lub przez komputer) wykorzystamy do sprawdzenia czy gra zakończyła się czyjąś wygraną, a także w jednej ze strategii komputera.

Wykorzystamy słownik, którego kluczami będą liczby, a wartościami lista par liczb wybranych przez danego gracza, których różnica arytmetyczna równa jest kluczowi.

Następnie, dla każdego klucza, korzystając z metodyki programowania dynamicznego znajdziemy długość najdłuższego ciągu o różniczy arytmetycznej równej kluczowi, spośród liczb wybranych przez użytkownika.

Pseudokod:

1. S <- Pusty słownik
2. (a_1, a_2, ..., a_n) <- liczby wybrane przez gracza
3. Dla i=1, 2, ..., n:
    - 3.1. Dla j=i+1, i+2, ..., n:
        - 3.1.1. r = $a_j – a_i$
        - 3.1.2. Do S[r] dodaj parę (a_i, a_j)
4. M = -1
5. Dla każdego k będącego kluczem S:
    - 5.1. T <- tablica jedynek o długości n
    - 5.2. Dla każdej pary (p,q) z S[k]:
        - 5.2.1. T[q] = T[p] + 1
    - 5.3. Jeżeli max(T) > M to:
        - 5.3.1. M <- max(T)
6. Najdłuższy ciąg arytmetyczny wynosi M

Złożoność wynosi **O(n^2)**. Drobne modyfikacje algorytmu nie zmieniające jego złożoności pozwolą nam, oprócz długości M najdłuższego ciągu, uzyskać jego różnicę arytmetyczną a także liczby wchodzące w jego skład.

Strategie gry komputera

Gracz na początku gry ma możliwość wyboru poziomu gry (łatwy/trudny). W zależności od tego, stosowana jest odpowiednia strategia komputera.

Poziom łatwy – strategia losowa

Komputer w swojej rundzie wybiera dwie liczby spośród jeszcze nie wybranych w sposób losowy.
Mając do wyboru dwie liczby wybrane przez gracza, spośród nich wybiera jedną również w sposób losowy.

Poziom trudny – strategia zachłanna

Komputer w swojej rundzie pierwszą liczbę wybiera taką, której pokolorowanie pozwoli uzyskać najdłuższy ciąg arytmetyczny. Sprawdzamy to za pomocą algorytmu na wyszukiwanie najdłuższego ciągu arytmetycznego przedstawionego wcześniej, do zbioru pokolorowanych liczb dodając niewybraną jeszcze przez nikogo liczbę. W przypadku kilku możliwości wybieramy dowolną liczbę.

Następnie druga liczba w rundzie wybierana jest w taki sam sposób, zakładając że liczba pierwsza została wybrana przez przeciwnika.

Złożoność takiej strategii wynosi **$O(n^3)$**.

W przypadku odwróconej rundy, komputer wybiera jedną liczbę spośród dwóch w taki sam sposób porównując która z nich pozwoli uzyskać najdłuższy ciąg.

Poziom bardzo trudny – ulepszona strategia zachłanna

Strategia działa tak jak powyższa strategia zachłanna, z następującymi modyfikacjami:

- W rundzie gdzie komputer wybiera dwa liczby, dla każdej liczby oprócz sprawdzenia długości **p** najdłuższego ciągu jaki uzyskamy wybierając daną liczbę, w ten sam sposób obliczana jest długość **q** najdłuższego ciągu jaką osiągnąłby nasz przeciwnik.
  Komputer wybiera liczbę, której odpowiada największa wartość **p**, spośród liczb dla których **p**>**q**.
  W przypadku braku takich liczb, wybieramy dowolną liczbę dla której **p**=**q**.
  Jeśli i takich liczb nie ma, wybieramy liczbę, której odpowiada najmniejsza wartość **q**.
- Taką samą zasadę jak powyższa, wprowadzamy do wyboru jednej liczby z dwóch w rundzie odwróconej.

Bibliografia

1. https://pl.wikipedia.org/wiki/Twierdzenie_Szemer%C3%A9diego
2. https://en.wikipedia.org/wiki/Szemer%C3%A9di's_theorem
3. https://codercareer.blogspot.com/2014/03/no-53-longest-arithmetic-sequence.html