

АЛГОРИТМИ ХАНОЙСЬКОЇ ВЕЖІ: АНАЛІЗ ТА ОПТИМІЗАЦІЯ В КОНТЕКСТІ РОЗВ'ЯЗАННЯ КОМБІНАТОРНИХ ПРОБЛЕМ

Сучасний розвиток обчислювальної науки та інформаційних технологій визначає актуальність вивчення та оптимізації алгоритмів Ханойської вежі, що є ключовим елементом в теорії обчислювальної складності. Алгоритми Ханойської [1] вежі є фундаментальними для обчислювальної науки та технологій. Вони використовуються для розв'язання багатьох комбінаторних задач, таких як сортування, пошуки та розпізнавання.

Ця теза присвячена вивченню та оптимізації алгоритмів Ханойської вежі. Дослідження спрямоване на покращення якості та швидкодії вирішення комбінаторних задач за допомогою алгоритмів Ханойської вежі. Це важливо для розвитку обчислювальної математики та інформаційних технологій.

Лістинг коду:

```
...
void hanoi(int n, int i, int k){
    if (n == 0){
        printf("\n\tMove disk 1 from pin %d to %d", i, k);
    }
    else{
        int tmp = 6 - i - k;
        hanoi(n - 1, i, tmp);
        printf("\n\tMove disk %d from pin %d to %d", n, i, k);
        hanoi(n - 1, tmp, k);
    }
}
int main(){
    system("chcp 1251");
    system("cls");
    int choice_1, choice_2, choice_3;
    printf("\n\tВведіть розмір вежі: ");
    scanf_s("%d", &choice_1);
    printf("\n\tВведіть номер стовпця якого перекладаємо: ");
    scanf_s("%d", &choice_2);
    printf("\n\tВведіть номер стовпця який перекладаємо: ");
    scanf_s("%d", &choice_3);
    hanoi(choice_1, choice_2, choice_3);
    printf("\n");
    return 0;
}
```

Алгоритм Ханойської вежі включає переміщення дисків різного розміру між трьома стовпцями: початковим, допоміжним та цільовим. Основна ідея – переміщати диски один за одним, дотримуючись двох основних правил: диск завжди може бути розміщений лише на диск меншого розміру, і переміщення відбувається тільки між трьома стовпцями.

Рекурсивний алгоритм Ханойської вежі включає три основні етапи:

- Переміщення (n-1) дисків з початкового стовпця на допоміжний.
- Переміщення найбільшого диска з початкового стовпця на цільовий.
- Переміщення (n-1) дисків з допоміжного стовпця на цільовий.

Цей процес повторюється рекурсивно для кожної підзадачі, поки всі диски не будуть переміщені на цільовий стовпець. Алгоритм Ханойської вежі є ефективним способом вивчення рекурсії та стратегій алгоритмізації.

Існують різні способи оптимізації алгоритму Ханойської вежі. Ось деякі з них:

Ітеративний метод. З допомогою цього метода можна уникнути накопичення великої кількості рекурсивних викликів, що може бути важливо для оптимізації часу виконання.

Паралельне виконання. Можливість паралельного виконання алгоритму за допомогою конкурентних або розподілених обчислень покращує швидкодію, особливо для великих обсягів даних.

Оптимізація для конкретних випадків. Враховуючи особливості задачі Ханойської вежі, можна розглянути оптимізації для конкретних випадків, наприклад, для задачі з меншою кількістю дисків.

Отже, оптимізація алгоритму Ханойської вежі може покращити його швидкодію та ефективність. Існують різні способи оптимізації, які можна використовувати окремо або в комбінації.

Список використаних джерел:

1. Алгоритм «Ханойська вежа» [Електронний ресурс] – Режим доступу до ресурсу: <https://ua5.org/osnprog/255-algoritm-khanojjska-vezha.html>
2. Ханойські вежі [Електронний ресурс] – Режим доступу до ресурсу: <https://www.eolymp.com/uk/problems/6187>