ДОДАТОК А

СЛАЙДИ ПРЕЗЕНТАЦІЇ

1

Мета роботи

Метою роботи є дослідження методів візуалізації алгоритмів та програмного коду для навчання та розвитку навичок програмування після здобуття базових знань.

2

Існуючі візуалізації

На поточний момент існує багато візуалізацій алгоритмів, починаючі із алгоритмів сортування або обходу графів та закінчуючи візуалізацією алгоритму самплінгу.

3

Причини для візуалізації алгоритмів

- Навчання
- Налагодження
- Розуміння
- Розвага

4

Глибина візуалізації алгоритмів

- Рівень 0 / чорна коробка – найпростіший випадок, тільки показує результат роботи алгоритму.
- Рівень 1 / сіра коробка – візуалізація кроків алгоритму
- Рівень 2 / біла коробка – показує внутрішній стан алгоритму у додаток до проміжних результатів

5

Об’єкт візуалізації

Візуалізація пов’язана з об’єктом, що візуалізуєтсья, найчастіше це структура даних: масив, граф...

Або також це може бути клітинний автомат.

6

Математична модель клітинних автоматів

Традиційна модель клітинних автоматів виглядає наступним чином:

$$\sigma_{i_1,i_2,\dots,i_n}(t+1) = \varphi(\sigma_{j_1,j_2,\dots,j_n} \mid j_m \in N(i_1,\ i_2,\dots,i_n))$$

де n – вимірність клітинного автомату,

t – момент часу,

φ – функція правила переходу,

$N(i_1,\ i_2,\dots,i_n)$ – деяка околиця точки $(i_1,\ i_2,\dots,i_n)$

7

Розширення моделі клітинних автоматів

$$\sigma_{i_1,i_2,\dots,i_n}(t+1) = \varphi(\sigma_{j_1,j_2,\dots,j_n} \mid j_m \in N(i_1,\ i_2,\dots,i_n), \sigma_{i_1,i_2,\dots,i_n}, s(f))$$

де n – вимірність клітинного автомату,

t – момент часу,

φ – функція правила переходу,

$N(i_1,\ i_2,\dots,i_n)$ – деяка околиця точки $(i_1,\ i_2,\dots,i_n)$,

$\sigma_{i_1,i_2,\dots,i_n}$ – стан самої точки $(i_1,\ i_2,\dots,i_n)$,

$s(f)$ – функція від загального стану поля

8

Розширення моделі клітинних автоматів

$$\sigma_{i_1,i_2,\dots,i_n}(t+1) = \begin{cases} \omega_1\left(\sigma_{j_1,j_2,\dots,j_n} \middle| j_m \in N(i_1,i_2,\dots,i_n), s(f)\right), \sigma_{i_1,i_2,\dots,i_n}(t) = 0 \\ \omega_2\left(\sigma_{j_1,j_2,\dots,j_n} \middle| j_m \in N(i_1,i_2,\dots,i_n), s(f)\right), \sigma_{i_1,i_2,\dots,i_n}(t) = 1 \\ \dots \\ \omega_k\left(\sigma_{j_1,j_2,\dots,j_n} \middle| j_m \in N(i_1,i_2,\dots,i_n), s(f)\right), \sigma_{i_1,i_2,\dots,i_n}(t) = k \end{cases}$$

де n – вимірність клітинного автомату,
t – момент часу,
ω_n – функція правила переходу для відповідного стану самої клітинки,
$N(i_1, i_2, \dots, i_n)$ – деяка околиця точки $(i_1, i_2, \dots, i_n)$,
$\sigma_{i_1,i_2,\dots,i_n}$ – стан самої точки $(i_1, i_2, \dots, i_n)$,
$s(f)$ – функція від загального стану поля

9

Приклад завдання

Наприклад, можна зробити завдання зробити модифікацію гри “життя”, із додаванням впливу функції від загального стану поля, яка буде сприяти меншому росту популяції, коли на полі занадто багато клітинок і, навпаки, сприяти більшому росту при малої популяції.

$$s(f) = bias\left(count\left(alive(f)\right)\right);$$

$$bias(n) = \begin{cases} 1, & \frac{n}{count(f)} \le 0.3 \\ 0, & 0.3 < \frac{n}{count(f)} < 0.7 \\ -1, & \frac{n}{count(f)} \ge 0.7 \end{cases}$$

10

Програмна реалізація

У вікні завдання користувач пише код та настроює модель.

11

Програмна реалізація

Потім описані класи візуалізуються згідно з заданими параметрами

12

Висновки

У цій роботі було розглянуто різні засоби візуалізації алгоритмів на різних структурах даних з точки зору навчання програмуванню та розробці алгоритмів з їхньою допомогою.

Згідно з результатами було вирішено створити модифікацію клітинних автоматів, яка б була найбільше спрямована на розвиток навичок програмування при створенні нових правил та алгоритмів.