CSE3013 (컴퓨터공학 설계 및 실험 I)
PRJ-2 미로 프로젝트 1주차 결과 보고서

서강대학교 컴퓨터공학과 박수현 (20181634)

서강대학교 컴퓨터공학과

1 목적

미로 게임을 위한 알고리즘과 자료구조를 이해한다.

2 문제

2.1 완전 미로 생성 알고리즘

실제로 사용한 미로 생성 알고리즘은 예비 보고서에서 작성한 프림 알고리즘과는 다른 알고리즘을 사용하였다. 바로 백트래킹이다. 높이 h, 너비 w의 미로에서 백트래킹은 다음과 같이 작동한다.

```
BACKTRACK(x, y, h, w)
Calculate unvisited adjacent cells to (x, y)
Set visited state of (x, y) to TRUE
while there exists unvisited adjacent cells to (x, y)
    (nx, ny) = random unvisited adjacent cell to (x, y)
    Make path from (x, y) to (nx, ny)
    BACKTRACK(nx, ny, h, w)
    Recalculate unvisited adjacent cells to (x, y)
```

랜덤하게 결정한 (x, y)에서 백트래킹을 시작하면 된다.

자료구조는 가로 길을 나타내는 $h \times (w-1)$ 크기의 불 배열과 세로 길을 나타내는 $(h-1) \times w$ 크기의 불 배열로 표현하였다. 실제로는 각각 `horizontal_route`와 `vertical_route`라는 이름으로 구현되었는데, 미로에서 (x, y)에서 $(x, y-1)$로 갈 수 있다면 `vertical_route[y - 1][x]`가 `1`이고, 아닐 경우 `0`인 자료구조이다. 이 자료구조의 공간 복잡도는 $\mathcal{O}(h \times w)$이다.

인접한 셀의 개수를 계산하는 알고리즘은 4방향만 체크하면 되므로 $\mathcal{O}(1)$으로 충분하다. 백트래킹은 각 셀마다 인접한 미방문 셀을 방문하는데, 결국 모든 셀을 방문하게 되므로 시간 복잡도도 $\mathcal{O}(h \times w)$이다.

2.2 불완전 미로 생성 알고리즘

불완전 미로의 생성은 생성된 완전 미로에서 추가로 몇 갸의 벽을 없애 주는 것으로 가능하다. 모든 칸들을 랜덤하게 돌면서 옆 셀과 벽으로 막혀 있는 셀이 있다면 길을 만들어 주는 식으로 n번 반복한다. 자료구조는 그대로 사용하였다.

DELETE-WALLS($delete$, h, w)

$delete = \min\left(delete, \frac{\min(h,w)}{2}\right)$

while $delete > 0$

  $(x, y) =$ random cell in maze field

  $(nx, ny) =$ random adjacent cell to (x, y)

  if there is no direct path between (x, y) and (nx, ny)

    Make new direct path

    $delete = delete - 1$

정점이 hw개 존재하는 그래프의 최소 신장 트리에는 노드가 $hw-1$개 존재한다. 또한 미로의 모든 벽을 삭제하면 간선은 $h \times (w-1) + (h-1) \times w$개 존재한다. 따라서 완전 미로 상태에서 셀과 셀 사이의 벽은

$$\begin{aligned} & [h \times (w-1) + (h-1) \times w] - [hw-1] \\ = & hw - h - w + 1 \end{aligned}$$

개이다. 따라서 총 $h \times (w-1) + (h-1) \times w$개의 원래 존재하던 벽 중 $hw-h-w+1$개를 고르는 확률로 벽을 지우는 데 성공할 수 있다.

n개의 벽 중 k개의 벽이 남아 있을 때, 지우기에 성공하는 확률은 $\dfrac{k}{n}$이다. 따라서 이 때 처음으로 존재하는 벽을 지우기까지의 시도 횟수를 X라 하면 $X \sim \mathrm{Exp}\left(\dfrac{k}{n}\right)$이므로 이 때 $\mathrm{E}(X) = \dfrac{n}{k}$이다.

이 알고리즘에서 벽을 r개 지우도록 한다면 벽 r개를 지우는 데 필요한 시도 횟수의 기댓값은 n번째 조화수 $H_n = \displaystyle\sum_{k=1}^{n} \frac{1}{k}$라고 할 때

$$\begin{aligned} & \sum_{k=0}^{r-1} \frac{h \times (w-1) + (h-1) \times w}{hw-1-k} \\ = & \sum_{k=0}^{r-1} \frac{2hw-h-w}{hw-1-k} \\ = & (2hw-h-w)\left(H_{h \times w-1} - H_{h \times w-r-1}\right) \end{aligned}$$

이다.

한편 상수 $\gamma \approx 0.577$에 대해 $\lim_{n\to\infty}(H_n - \ln n) = \gamma$임이 알려져 있으므로, $hw = A \to \infty$일 때

$$
\begin{aligned}
&\lim_{A\to\infty}\sum_{k=0}^{r-1}\frac{h\times(w-1)+(h-1)\times w}{hw-1-k}\\
=&\lim_{A\to\infty}\left[(2A-h-w)\left(H_{A-1}-H_{A-r-1}\right)\right]\\
=&\lim_{A\to\infty}\left[(2A-h-w)\left(\ln(A-1)-\ln(A-r-1)\right)\right]\\
=&\lim_{A\to\infty}\left[(2A-h-w)\ln\frac{A-1}{A-r-1}\right]\\
=&\lim_{A\to\infty}\ln\left(\frac{A-1}{A-r-1}\right)^{2A-h-w}\\
=&\lim_{A\to\infty}\ln\left(1+\frac{r}{A-r-1}\right)^{2A-h-w}\\
=&\lim_{A\to\infty}\ln\left(1+\frac{r}{A-r-1}\right)^{2r\times\frac{A-r-1}{r}+2r+1-h-w}\\
=&\lim_{A\to\infty}\ln\left[\left[\left(1+\frac{r}{A-r-1}\right)^{\frac{A-r-1}{r}}\right]^{2r}\left(1+\frac{r}{A-r-1}\right)^{2r+1-h-w}\right]\\
=&\lim_{A\to\infty}\ln\left[\left(1+\frac{r}{A-r-1}\right)^{\frac{A-r-1}{r}}\right]^{2r}+\lim_{A\to\infty}\ln\left(1+\frac{r}{A-r-1}\right)^{2r+1-h-w}\\
=&\ln\lim_{A\to\infty}\left[\left(1+\frac{r}{A-r-1}\right)^{\frac{A-r-1}{r}}\right]^{2r}+(2r+1-h-w)\ln\lim_{A\to\infty}\left(1+\frac{r}{A-r-1}\right)\\
=&\ln e^{2r}+(2r+1-h-w)\ln 1\\
=&2r
\end{aligned}
$$

이고, 따라서 hw의 미로에서 벽을 r개 지운다면 평균적으로 $2r$번의 연산이 필요하다. 결론적으로 평균적인 경우 완전 미로에서 벽들을 더 지워 불완전 미로를 만드는 데 드는 시간 복잡도는 $\mathcal{O}(r)$으로 계산 가능하다. 불완전 미로를 처음부터 새로 만드는 경우를 생각한다면 완전 미로를 먼저 만들어야 하므로 전체 시간 복잡도는 $\mathcal{O}(h\times w+r)$이다.

자료구조는 동일한 것을 사용하였으므로 공간 복잡도도 마찬가지로 $\mathcal{O}(h\times w)$이다.