



국민대학교  
소프트웨어대학  
소프트웨어학부

# 알고리즘 설계 프로젝트

프로젝트 명	<i>Hidato Puzzle</i>
팀 명	<i>88</i>
문서 제목	제안서

Version	1.0
Date	2018-NOV-10

팀원	안은영 (팀장)
	박기훈
	김현석
	김용태
	황승애
	송성유
지도교수	최 준수 교수님

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>알고리즘</b>	<b>제안서</b>		
	<b>프로젝트 명</b>	Hidato Puzzle	
	<b>팀 명</b>	88	
	Confidential Restricted	Version 1.0	2018-NOV-10

#### CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 소프트웨어대학 소프트웨어학부 개설 교과목 알고리즘 수강 학생 중 프로젝트 "**Hidato Puzzle**"를 수행하는 팀 "**88**"의 팀원들의 자산입니다. 국민대학교 소프트웨어학부 및 팀 "**88**"의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

## 목 차

1	개요 .....	3
2	프로젝트의 목표 .....	3
3	요구사항 .....	3
4	해결 방안 .....	4
5	Risk Management Plan .....	5
6	프로젝트 팀 구성 및 역할 분담 .....	6

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>알고리즘</b>	<b>제안서</b>		
	<b>프로젝트 명</b>	Hidato Puzzle	
	<b>팀 명</b>	88	
	Confidential Restricted	Version 1.0	2018-NOV-10

## 1 개요

이스라엘 수학자인 Dr. Gyora M. Bened다는 논리 퍼즐 게임을 고안했다. 바로 Hidato Puzzle 게임이다. 이 게임은 수평, 수직 또는 대각선으로 연결되는 연속 번호로 그리드를 채우는 것이다. Hidato Puzzle에서 가장 작은 숫자와 가장 높은 숫자가 그리드에 주어진다. 플레이어에게 솔루션을 시작하는 방법을 지시하고 Hidato가 단일 솔루션을 갖도록 하기 위해 그리드에는 (가장 작은 값과 가장 높은 값 사이) 주어진 다른 번호도 있다. 잘 만들어진 Hidato Puzzle은 유일한 해답이 존재해야 한다. 그리고 Hidato Puzzle은 단순한 논리에 의해 발견될 수 있는 해결책을 가지고 있어야 한다.

## 2 프로젝트의 목표

Hidato Puzzle은 가장 낮은 숫자로부터 가장 높은 숫자로 수평, 수직, 대각선으로 연결되는 위치로 연속 번호를 매기면서 그리드를 채우는 게임이다. 이번 프로젝트에서는 이 Hidato Puzzle을 Generate 하고 Solve를 하는 프로그램을 구현하는 것이다. 이에 두개의 조로 나누어 Generator와 Solver로 나누어 프로그램 구현을 진행할 것이다.


## 3 요구사항

### 3.1 Generator

빈칸의 갯수  $n$ 개에 대해 1부터  $n$ 까지의 숫자가 가로, 세로, 대각선으로 연결되어 있는 정답 퍼즐을 생성해 주어야 한다. 그리고 그 생성된 정답 퍼즐에서, 여러개의 정답 퍼즐이 생성되지 않게 빈칸을 생성해 주어야 한다. 그리고 해당 퍼즐이 제대로 만들어 졌는지 확인을 해준다.

### 3.2 Solver

1부터 목표 숫자까지 떨어지지 않고 연결된 숫자를 그리드에 채워야 한다. 연결된 숫자는 수평, 수직, 대각선으로만 이동할 수 있다. 숫자를 이어 나갈 때 빈공간이 발생하여, 추후에 숫자를 쓸 공간이 없는 것을 방지해야 한다. 막혀 있는 공간으로는 숫자를 쓸 수 없다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>알고리즘</b>	<b>제안서</b>		
	<b>프로젝트 명</b>	Hidato Puzzle	
	<b>팀 명</b>	88	
	Confidential Restricted	Version 1.0	2018-NOV-10

## 4 해결 방안

### 4.1 Generator

위의 요구 사항을 단계별로 나누면 첫 번째로 길(정답 퍼즐)을 생성하고, 생성된 길을 토대로 적절한 개수의 빈칸을 뚫어주는 일이다. 이를 길 생성, 빈칸 생성 2단계의 알고리즘으로 분류해, 순차적으로 실행시킨다.

#### 4.1.1 정답 퍼즐 생성 알고리즘

우선 랜덤한 locaton에  $n=1$ 의 값 부여, 갈 수 있는 모든 공간에 대해  $n+1$ 의 값을 부여하며 재귀적으로 실행하고, 길이 막히면 되돌아가게 하여 모든 칸이 채워질 때까지 반복한다. 만약 모든 경우의 수를 돌았을 때 만족하지 않는다면 모든 값을 삭제 후 맵을 새로 받아와 처음의 위치를 제외한 랜덤한 위치에  $n=1$  을 다시 생성 후, 위의 알고리즘 반복한다.

#### 4.1.2 빈칸 생성 알고리즘 -

생성된 퍼즐에 대해 한가지의 해답을 유지하면서 숫자를 비워놓기 위해 랜덤한 숫자를 지울 때마다 여러 개의 해답이 생성되는지를 확인하고 ,만약 여러 개의 해답이 생성된다면 그 숫자는 삭제하지 않고 유지한 채, 다음 숫자에 대한 빈칸 생성 알고리즘을  $n-2$  번(1과  $n$ 제외) 반복한다. 빈칸의 개수가 너무 작거나 많을 경우, 알고리즘을 재실행 (빈칸의 개수 폭을 단계적으로 부여하여 난이도 조절까지 가능) 한다.

### 4.2 Sovler

bfs(너비 우선 탐색) 알고리즘을 이용하여 문제를 해결한다. 출발 지점으로부터 갈 수 있는 정점을 queue에 넣고 첫번째 정점으로부터 탐색을 시작한다. 탐색이 종료가 되었을 때 목표 지점에 도달 하지 못했다면, queue에서 빼고 queue에 그 다음 정점으로 다시 탐색을 시작한다. 이 작업을 반복하다 만약 탐색이 정상적으로 되었다면 입력된 결과값을 저장하고 종료한다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>알고리즘</b>	<b>제안서</b>		
	<b>프로젝트 명</b>	Hidato Puzzle	
	<b>팀 명</b>	88	
	Confidential Restricted	Version 1.0	2018-NOV-10

## 5 Risk Management Plan

프로젝트 수행 중 발생할 가능성이 있는 위험 요소를 미리 예측하여 나열한다. 해당 위험 요소가 발생할 경우 대처할 방안에 대해 명시한다.

Items	Description	Management Plan
미완성된 Generate 발생	1에서 n까지의 모든 퍼즐이 완성되지 않았는데 모든 경우에 대해서 길이 끊김	새로운 시작점(1) 지정 후 퍼즐 생성 재시작
여러 개의 정답이 존재하는 Puzzle 이 generate 될 경우	빈칸을 비워서 퍼즐을 완성하였는데, 퍼즐에 대한 정답이 여러 개가 완성됨	여러개의 정답퍼즐이 완성되지 않도록 빈칸을 생성할 때 마다 여러개의 정답이 생성되는지 체크
퍼즐 생성의 성능	퍼즐을 생성하는 생성 시간이 지나치게 오래 걸린다. 또는 필요 이상의 자원을 사용한다.	더 좋은 컴퓨터를 사용한다 또는, 일정 부분 단순해 지더라도, 조금 더 단순한 알고리즘을 사용한다.
퍼즐 난이도 선정의 어려움	생성된 퍼즐의 난이도가 너무 쉽거나 너무 어렵다.	빈칸 생성 알고리즘을 다시 실행하여 새로운 퍼즐을 완성시키고, 일정 횟수 이상의 이 작업에 대해서도 걱정한 난이도의 빈칸이 생성되지 않을 경우, 정답 퍼즐 생성 알고리즘을 다시 실행하여 새로운 퍼즐 생성

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>알고리즘</b>	<b>제안서</b>		
	<b>프로젝트 명</b>	Hidato Puzzle	
	<b>팀 명</b>	88	
	Confidential Restricted	Version 1.0	2018-NOV-10

## 6 프로젝트 팀 구성 및 역할 분담

프로젝트에 참여하는 멤버의 역할을 구체적으로 명시한다.

이름	역할
박기훈	Hidato Puzzle 을 Generate 하는 알고리즘을 구현
송성유	유일한 답인지 Check 하는 알고리즘을 구현
황승애	Generate 알고리즘 및 Check 알고리즘 검사
안은영	(조장)선정된 답과 Solve 된 답이 일치하는지 확인하는 알고리즘 구현
김용태	Hidato Puzzle 에 사용될 bfs(너비우선탐색)알고리즘 구현
김현석	Hidato Puzzle 에 사용될 bfs(너비우선탐색)알고리즘 구현