AI 프로그래밍

#HOMEWORK 09



학번: 201724461

이름: 류지환

번외) HW08 제출물 수정 사항

1. problem의 멤버 함수였던 displaySetting(msg)을 problem 외부 함수로 변경.
2. 외부 함수 중 Mutate 계열 함수,

first-choice(n): {randomMutant(current, p), mutate(current, i, d, data)},

first-choice(tsp): { randomMutant(current, p)},

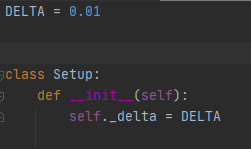
steepest-ascent(n): { mutants(current, p), mutate(current, i, d, p),bestOf(neighbors, p)}, steepest-ascent(tsp): { mutants(current, p), bestOf(neighbors, p) },

gradient-decent(n): { gradient(p, current, valueC) -> gradientMutate(current, valueC),

gradientMutant(p, grd, current) -> gradientMutant(grd, current) }

를 각 소속 problem 계열의 멤버 함수로 포함시킴(ex: first-choice(n)의 외부 함수는 numeric의 멤버 함수로 포함).

1. Setup class:
   1. 함수 개요:



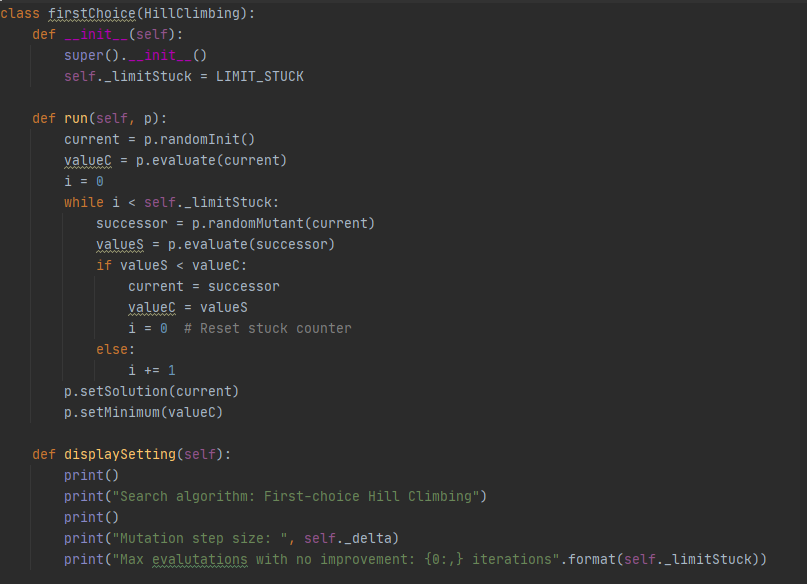
* 1. Grobal 변수: DELTA(0.01)
  2. Class화 고려 사항:
     1. Problem class와 HillClimbing의 super class로 두 class에서 중복되게 사용되는 DELTA를 멤버 변수로 포함.

1. HillClimbing class:
   1. 함수 개요:



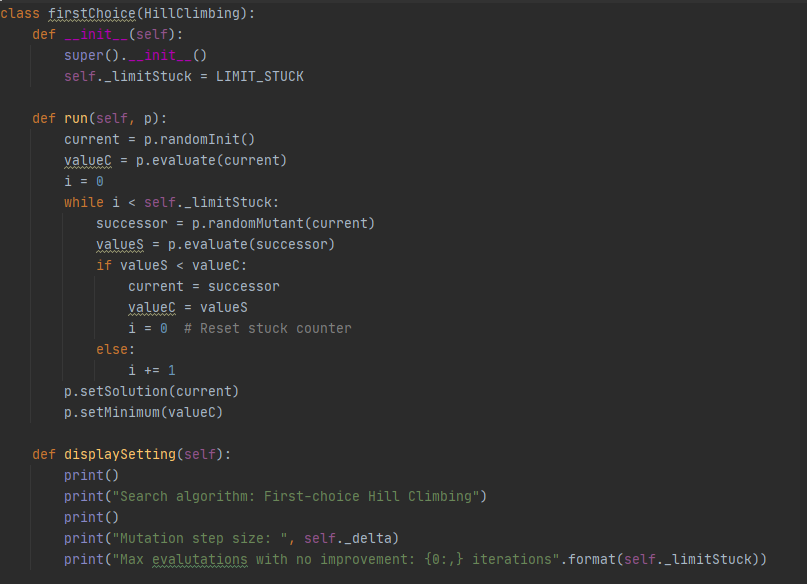
* 1. grobal 변수: LIMIT\_STUCK(100), EPSILON(0.0001)
  2. Class화 고려 사항:
     1. 하위 클래스(firstChoice, steepestAscent, gradientDecent)의 interface.

1. firstChoice class:
   1. 함수 개요:



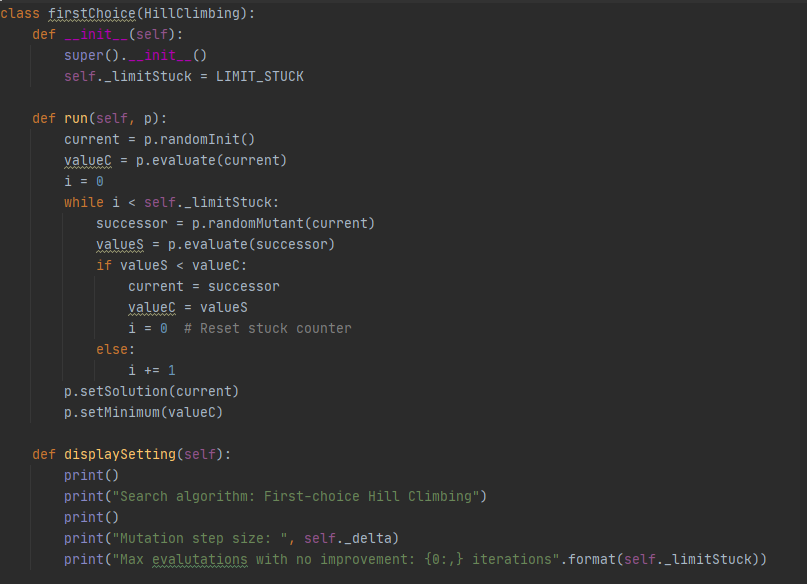
* 1. Class화 고려사항:
     1. Firs-choice 알고리즘의 LIMIT\_STUCK을 멤버 변수 limitStuck으로 포함.
     2. 기존 HW08에서 외부 함수였던 firstChoice(p)를 firstChoice class의 내부 함수 run(self, p)로 구현
     3. 기존 HW08에서 외부 함수였던 disPlaySetting()을 frist-choice 출력문 양식에 맞춰 수정 후 firstChoice class의 내부 함수로 구현

1. steepestAscent class:
   1. 함수 개요:



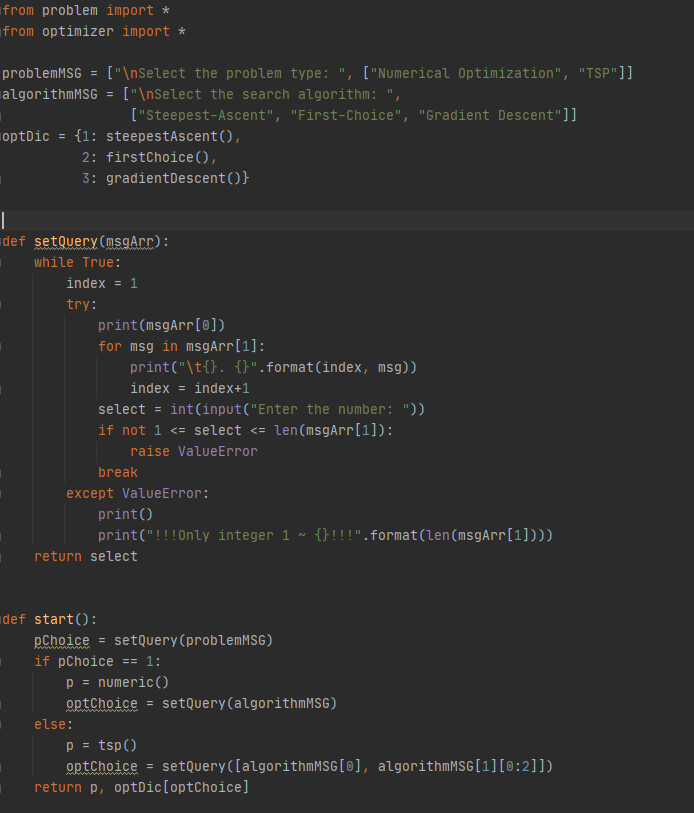
* 1. Class화 고려사항:
     1. 기존 HW08에서 외부 함수였던 steepestAscent(p)를 steepestAscent class의 내부 함수 run(self, p)로 구현
     2. 기존 HW08에서 외부 함수였던 disPlaySetting()을 steepestAscent 출력문 양식에 맞춰 수정 후 steepestAscent class의 내부 함수로 구현

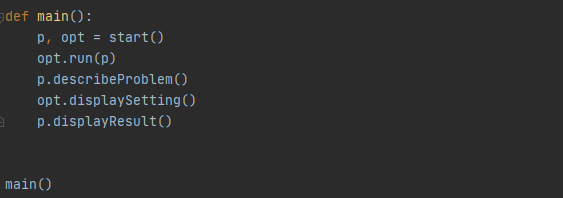
1. gradientDescentclass:
   1. 함수 개요:



* 1. Class화 고려사항:
     1. gradientDescent 알고리즘의 EPSILON을 멤버 변수 epsilon으로 포함.
     2. 기존 HW08에서 외부 함수였던 gradientDescent(p)를 gradientDescent class의 내부 함수 run(self, p)로 구현
     3. 기존 HW08에서 외부 함수였던 disPlaySetting()을 gradientDescent 출력문 양식에 맞춰 수정 후 gradientDescent class의 내부 함수로 구현.

1. main def:
   1. 함수 개요:





* 1. Global 변수: problemMSG[2], algorithmMSG[2], optDic
  2. 함수 구성 시 고려사항:
     1. 출력 양식을 problem과 algorithm으로 나누어 각각 list로 저장.
     2. 사용자가 입력한 쿼리에 맞는 algorithm을 생성해야 함으로 이를 optDic으로 구현.
     3. SetQuery(msgArr)를 거쳐 사용자가 원하는 쿼리를 입력 받음.
     4. SetQuery(msgArr) 진행 중 제시된 예시 이외의 값을 사용자가 입력할 경우 ValueError를 발생시키고 경고문 출력.
     5. 위 사항들을 start 함수에서 동작하도록 하고 사용자가 선택한 problem과 algorithm을 main 함수의 p와 opt에 저장.
     6. p와 opt의 멤버 함수들을 사용해 문제 해결.