

Para que os alunos possam ir desenvolvendo o trabalho à medida que a matéria vai sendo dada, deve ser seguido o seguinte trajeto:

1. Realizar o predicado **ligaBid(Casa1,Casa2)** que produz por retrocesso as duas possibilidades de se percorrer a ligação entre as casas **Casa1** e **Casa2**, onde **Casa1** e **Casa2** têm a forma **casa(X,Y)**.
2. Assumindo a representação de estado que achar mais conveniente para este problema, construa o predicado **podeJuntar(NovoEstado,PercursoAnterior,NovoPercurso)** que falha se **PercursoAnterior** já incluir a casa corrente em **NovoEstado**; caso contrário, junta essa nova casa no fim de **PercursoAnterior**, produzindo **NovoPercurso**.
3. (Para quem não fizer a parte optativa) Construa o predicado **novoEstado(+EstadoCorrente,-NovoEstado)** que, por retrocesso, gera os estados sucessores de **EstadoCorrente**.
4. (Só para quem faz a parte optativa) Construa o predicado **novoEstadoHC(+EstadoCorrente,-NovoEstado,-H)** que, por retrocesso, gera os estados sucessores de **EstadoCorrente** associados a uma estimativa do custo do percurso (**H**) para cada um deles.
5. Construa o predicado **resolverLabirinto(+Casa1,+Casa2,+Dmax,-Percurso/-Dist)** que instancia **Percurso** e **Dist**, respetivamente, com o percurso entre os lugares indicados e a distância (número de movimentos entre casas) total percorrida entre as casas **Casa1** e **Casa2**, onde **Casa1** e **Casa2** têm a forma **casa(X,Y)**. A solução não deve exceder a distância passada em **Dmax**. Para quem fizer a parte optativa, este predicado deve usar o algoritmo de pesquisa *hill climbing* e os predicados **podeJuntar** e **novoEstadoHC** desenvolvidos anteriormente; os restantes alunos deverão usar o algoritmo de pesquisa em profundidade e os predicados **podeJuntar** e **novoEstado** desenvolvidos anteriormente.
Nesta implementação **Dmax** e **Dist** podem não ser considerados.
Deve também construir o predicado **final** usado no algoritmo de pesquisa.
6. Construa o predicado **resolver(Xp,Yp,Xc,Yc,P)** usando os que construiu anteriormente. Se no ponto 5 considerou os parâmetros **Dmax** e **Dist**, assumo que **Dmax** é indicado com uma cláusula **dMax(+Valor)** e o valor de distância calculado pelo predicado **resolverLabirint** no parâmetro **Dist** deve ser usado para acrescentar à base de dados do Prolog uma cláusula **dPerc(Dist)**.