



03-01-2018

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Relatório do Projeto Castings TV

Parte 2

Algoritmos e Estruturas de Dados
Docente Ana Paula Rocha

Leonardo Gomes Capozzi – up201503708@fe.up.pt
Ricardo Carvalho – up201503717@fe.up.pt
Verónica Fradique – up201506440@fe.up.pt

Índice

Descrição do projeto	2
Introdução	3
Descrição da Implementação	4
Árvore Binária de Pesquisa	4
Fila de Prioridade	4
Tabela de Dispersão	4
Casos de Utilização do Castings TV	5
Conclusão.....	6

Descrição do projeto

Este projeto consiste numa adaptação do primeiro projeto “Castings TV” da unidade curricular de Algoritmos e Estruturas de Dados recorrendo ao uso de estruturas de dados como árvores de pesquisa binária, filas de prioridade e ainda tabelas de dispersão.

Para a realização deste projeto grupo baseou-se no enunciado seguinte:

“Para facilitar a criação de sessões, os candidatos devem poder ser ordenados por género de arte performativa e, dentro de cada uma delas, pela idade dos candidatos. Para isso, guarde os candidatos numa árvore binária de pesquisa e implemente funcionalidades para identificar todos os candidatos de um dado género de arte performativa. Deve também suportar a alteração dos atributos do candidato bem como operações de inserção e remoção.

São realizadas entrevistas aos candidatos de preparação para as sessões, não só quando é a primeira sessão desse candidato mas também quando já realizou outras sessões de casting no passado. As entrevistas são organizadas por género de arte performativa. Use uma fila de prioridade para guardar os candidatos ordenada pela data da sessão mais recente que realizou em cada género de arte performativa (por exemplo, a sessão mais recente realizada em dança e a sessão mais recente em canto). Implemente também o registo de novas participações em sessão, atualizando a fila de prioridade respetiva.

Para conciliar as sessões com a agenda profissional dos candidatos, é-lhes pedido para indicarem os períodos em que estão indisponíveis para realizarem sessões. Os candidatos que estão indisponíveis, são transferidos para uma tabela de dispersão, sendo registado a razão da indisponibilidade. Quando terminar esse período de indisponibilidade, os candidatos voltam para a estrutura de dados onde estavam originalmente guardados. Por vezes, os candidatos desistem, permanecendo assim nesta tabela.”

Introdução

Este relatório encontra-se dividido em três secções, sendo elas sucintamente descritas em seguida:

- ❖ Descrição da Implementação: nesta secção aborda-se a solução que o grupo implementou para cumprir o objetivo pedido, de forma explícita;
- ❖ Casos de Utilização do Castings Tv: nesta secção descrevem-se as funcionalidade com que o programa conta;
- ❖ Conclusão: nesta secção descrevem-se as as principais dificuldades encontradas durante o desenvolvimento do projeto e ainda o resumo das contribuições de cada um dos membros para o projeto.

Descrição da Implementação

Neste projeto foram implementadas uma árvore binária de pesquisa, filas de prioridade e uma tabela de dispersão que serão seguidamente descritas.

De forma a facilitar a criação de sessões, os candidatos, inicialmente guardados num vetor, foram separados entre a árvore binária de pesquisa e a tabela de dispersão. Os candidatos guardados na árvore são os considerados válidos e, portanto, os que podem ser adicionados a sessões. Na tabela de dispersão foram guardados os candidatos considerados inválidos, que se encontram indisponíveis para participar em sessões.

Para facilitar a marcação de entrevistas de preparação para as sessões, referências para os candidatos foram guardadas em filas de prioridade, tal como abaixo descrito.

Árvore Binária de Pesquisa

A árvore binária de pesquisa foi utilizada para guardar os candidatos considerados válidos que estão disponíveis para concurso, sendo ordenados por forma a facilitar a sua adição a sessões. Assim foi definido o operador ‘menor’ da classe Candidato, usando o algoritmo que havia sido previamente usado na ordenação de candidatos, de forma a que estes ficassem ordenados alfabeticamente pela arte performativa e, dentro da mesma arte, crescentemente por idade. A árvore foi implementada nos ficheiros Empresa.hpp e Empresa.cpp.

Relativamente ao trabalho anterior, esta implementação apenas requereu a adaptação de funções que acediam ao vetor candidatos para acederem corretamente aos candidatos enquanto árvore binária de pesquisa.

Fila de Prioridade

As filas de prioridade são necessárias para apoio à realização de entrevistas nos castings. Cada fila de prioridade guarda os candidatos ordenados pela data da sessão mais recente em que estão inscritos e por género de arte performativa. A cada sessão em que o candidato é inscrito procede-se à atualização da fila de prioridades da respetiva arte para inserir este candidato.

Assim sendo, as filas encontram-se nos ficheiros Empresa.hpp e Empresa.cpp.

No ficheiro Empresa.hpp foi criada uma classe comp_candidatos e foi definido o operador() em que se comparam dois candidatos e se retorna o que está inscrito na sessão com data mais recente.

São também criadas três funções para trabalhar com as filas de prioridade: atualiza_candidato_pq, get_PQ e ainda adiciona_candidato_pq. A primeira função faz a atualização do candidato para a sua nova sessão e é chamada quando se adiciona um candidato a uma sessão. A segunda função apenas retorna a fila de prioridade de uma certa arte. Por sua vez, a terceira função adiciona o candidato passado como argumento da função à fila de prioridades da respetiva arte.

Tabela de Dispersão

Na tabela de dispersão são guardados os candidatos indisponíveis para participar em sessões, sendo estes considerados como inválidos. Quando um candidato é definido como inválido, este é retirado da árvore binária de pesquisa e adicionado na tabela de dispersão e, quando o candidato está outra vez disponível, é retirado da tabela e colocado de volta na árvore. A função de dispersão usada utiliza o número único de inscrição do candidato para calcular a posição deste dentro da tabela.

Casos de Utilização do Castings TV

Relativamente ao projeto anterior foi alterada a funcionalidade de ordenação de candidatos e foram adicionadas funcionalidades relativas às entrevistas de preparação para sessões e validade de candidatos.

No caso da ordenação de candidatos, como estes foram guardados numa árvore binária de pesquisa, a ordenação dos candidatos foi fixada, tendo sido retirada a opção de alterar a ordenação dos candidatos.

No que concerne às entrevistas foi disponibilizada uma opção que, dada uma arte performativa, mostra no ecrã os candidatos pela ordem de entrevista.

Relativamente à validade de candidatos, a opção de eliminação de um candidato foi substituída por indisponibilização do candidato sendo pedida uma razão para a sua indisponibilidade e tendo sido acrescentada a opção de marcar candidato como disponível, ou seja, validar o candidato.

Conclusão

O projeto encontra-se a cumprir as funcionalidades pretendidas e de forma eficiente. Para além disso conta com mecanismos de controlo de erros de inputs.

O grupo conclui que as principais dificuldades foram a adaptação do código às novas estruturas pretendidas e ainda a interpretação e adaptação do enunciado ao projeto que já estava criado.

No que diz respeito à avaliação do grupo conclui-se que todos os elementos despenderam do mesmo tempo e empenho na realização do mesmo.

Em suma, o projeto constituiu uma mais valia na compreensão das estruturas de dados utilizadas e ainda a clarificação das operações que se podem executar usando as mesmas.