|  |
| --- |
| #include <stdlib.h> |
|  | #include <stdio.h> |
|  | #include <stdbool.h> |
|  | #include <string.h> |
|  |  |
|  | #define MAX\_WORK\_COUNT 20 |
|  |  |
|  | #define WORK\_COST 0 |
|  | #define WORK\_DDL 1 |
|  | #define WORK\_VALUE 2 |
|  |  |
|  | #define STATE\_DDL\_OK 1 |
|  | #define STATE\_DDL\_EXECEED 2 |
|  |  |
|  | typedef struct DeployState { |
|  | int SelectedWorks[MAX\_WORK\_COUNT]; |
|  | int SelectedWorkCount; |
|  | int Cost; |
|  | int Value; |
|  | } DeployState, \*pDeployState; |
|  |  |
|  | static int list[][3] = { |
|  | //Cost ddl Value |
|  | 0, 0, 0, |
|  | 4, 10, 2, // 1 |
|  | 3, 8, 3, // 2 |
|  | 2, 4, 3, // 3 |
|  | 1, 2, 6 // 4 |
|  | }; |
|  |  |
|  | static DeployState maxDeployState = { 0 }; |
|  |  |
|  | int refreshDeployStateAndCheckDDL(pDeployState pDeployState) { |
|  | pDeployState->Cost = 0; |
|  | pDeployState->Value = 0; |
|  |  |
|  | for (int i = 0; i < pDeployState->SelectedWorkCount; i++) { |
|  | if (pDeployState->SelectedWorks[i] == 0) { |
|  | break; |
|  | } |
|  |  |
|  | pDeployState->Cost += list[pDeployState->SelectedWorks[i]][WORK\_COST]; |
|  | pDeployState->Value += list[pDeployState->SelectedWorks[i]][WORK\_VALUE]; |
|  |  |
|  | if (pDeployState->Cost > list[pDeployState->SelectedWorks[i]][WORK\_DDL]) { |
|  | return STATE\_DDL\_EXECEED; |
|  | } |
|  | } |
|  |  |
|  | return STATE\_DDL\_OK; |
|  | } |
|  |  |
|  | int copyStructDeployState(pDeployState src, pDeployState dst) { |
|  | for (int i = 0; i < MAX\_WORK\_COUNT; i++) |
|  | dst->SelectedWorks[i] = src->SelectedWorks[i]; |
|  | dst->Cost = src->Cost; |
|  | dst->Value = src->Value; |
|  | dst->SelectedWorkCount = src->SelectedWorkCount; |
|  |  |
|  | return 0; |
|  | } |
|  |  |
|  | int getDeployStateWorkCount(pDeployState pDeployState) { |
|  | // for (int i = 0; i < MAX\_WORK\_COUNT; i++) { |
|  | // if (pDeployState->SelectedWorks[i] == 0) { |
|  | // return i; |
|  | // } |
|  | // } |
|  | // return MAX\_WORK\_COUNT; |
|  | return pDeployState->SelectedWorkCount; |
|  | } |
|  |  |
|  | int dfsGetMaxValue(pDeployState pDeployStateCurrnet) { |
|  | // Recalculate Costs, Check DDL |
|  | if (refreshDeployStateAndCheckDDL(pDeployStateCurrnet) & STATE\_DDL\_EXECEED) { |
|  | return STATE\_DDL\_EXECEED; |
|  | } |
|  |  |
|  | // Compare Max |
|  | if (pDeployStateCurrnet->Value > maxDeployState.Value) { |
|  | copyStructDeployState(pDeployStateCurrnet, &maxDeployState); |
|  | } |
|  |  |
|  | // going on dfs |
|  | DeployState DeployStateNew; |
|  | copyStructDeployState(pDeployStateCurrnet, &DeployStateNew); |
|  | DeployStateNew.SelectedWorkCount++; |
|  |  |
|  | DeployStateNew.SelectedWorks[DeployStateNew.SelectedWorkCount - 1] = 1; |
|  | dfsGetMaxValue(&DeployStateNew); |
|  |  |
|  | DeployStateNew.SelectedWorks[DeployStateNew.SelectedWorkCount - 1] = 2; |
|  | dfsGetMaxValue(&DeployStateNew); |
|  |  |
|  | DeployStateNew.SelectedWorks[DeployStateNew.SelectedWorkCount - 1] = 3; |
|  | dfsGetMaxValue(&DeployStateNew); |
|  |  |
|  | DeployStateNew.SelectedWorks[DeployStateNew.SelectedWorkCount - 1] = 4; |
|  | dfsGetMaxValue(&DeployStateNew); |
|  |  |
|  | return STATE\_DDL\_OK; |
|  | } |
|  |  |
|  | int main(int argc, char \*argv[]) { |
|  | DeployState DeployStateInit = { 0 }; |
|  |  |
|  | dfsGetMaxValue(&DeployStateInit); |
|  |  |
|  | printf\_s("%d", maxDeployState.Value); |
|  |  |
|  | getchar(); |
|  | return 0; |
|  | } |