

課題4 PERTによるクリティカルパスの抽出

下の表に示す作業からなるプロジェクトについて、クリティカルパスを求める。以下の手順に従って結果を示せ。

表：作業の依存関係		
作業名	必要日数	前提作業
A	2	なし
B	3	なし
C	2	A
D	1	B
E	3	C
F	2	B, C
G	1	D
H	2	F, G
完了	–	E, H

- このプロジェクトのPERT用アローダイアグラムを示し、対応するグラフデータファイルを作成せよ。ただし、アローダイアグラムのプロジェクト開始頂点の番号は「1」にすること。
- 全ての頂点について最早結合点時刻を算出する関数「`calc_earliest_node_times`」と、最遅結合点時刻を算出する関数「`calc_latest_node_times`」を実装せよ。
- クリティカルパスをパスに沿った頂点リストを求める関数「`trace_critical_path`」を実装し、実行結果を示せ。ただし、複数の経路がある場合には、いずれかの経路を一つだけ示せば良い。以下のアルゴリズム概略を参考にせよ。
 - 開始頂点番号： $i \leftarrow 1$ とする。
 - 作業列を保存するリスト： $P \leftarrow \{\}$ （空にする）とする。
 - リスト P の末尾に v_i を追加する。
 - v_i の最早結合点時刻を t_i^E とする。
 - v_i に接続している次の全頂点 v_j について
 - v_j がなければ終了。
 - v_j の最遅結合点時刻を t_j^L とする。
 - v_j から v_i に要する作業時間を $t_{i,j}$ とする。
 - t_j^L と $t_i^E + t_{i,j}$ を比較する。
 - 一致する ($t_j^L = t_i^E + t_{i,j}$) 場合： $i \leftarrow j$ に置き換えて (3c) へ。
 - 一致しない場合：次の v_j について調べる。
 - リスト P に保存されている頂点列がクリティカルパスを表す。

■