Übung 09

21.06.22 13:47

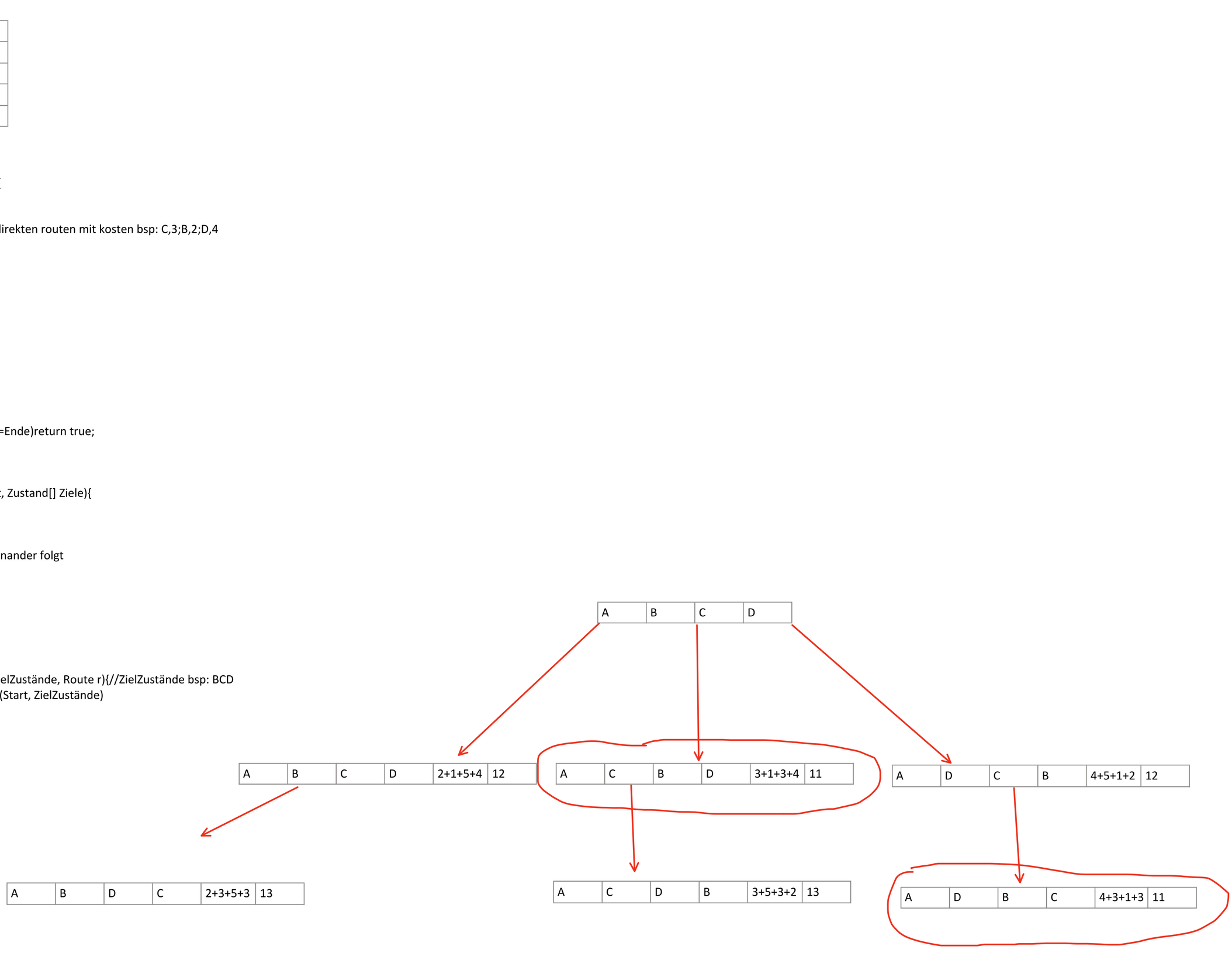
Backtracking: TSP

	A	B	C	D
A	0	2	3	4
B	2	0	1	3
C	3	1	0	5
D	4	3	5	0

Pseudocode:

```
Object Zustand(Char c, char[] z, int[] kosten){

	name = c;
	nachbarn[][]={z.length}; //Tuples von direkten routen mit kosten bsp: C,3;B,2;D,4
	for(i=0;i<nachbarn.length;i++){
		nachbarn[i][0]=z[i];
		nachbarn[i][1]=k[i];
		}
}

Object Route(Zustand[] z){
	Zustand[] z = z //Route
	calcRoute()//berechnetKosten
}

function RouteValide(Route r, Start, Ziel){
	if(routeEnthältAlleZustände && Start==Ende)return true;
	return false
}

function genereiereAlleRouten(Zustand Start, Zustand[] Ziele){
	Route[Ziele.len]= res
	for(i in Zustand){
	   res[i].Zustand[0]=Start
	//generiere Route nicht wenn CD aufeinander folgt
	   ...
	   res[i].Zustand[res.len]=Start
	}
	return res;
}


function backtack(Zustand Start, Zustand[] ZielZustände, Route r){//ZielZustände bsp: BCD
	Route[] routen= genereiereAlleRouten(Start, ZielZustände)
	Short[routen] res;
	for (route in routen){
		res.add(route.calcRoute())
		}
	return getKleinsterWertAusListe(res);
}
```

Nr. 2
Sei A=s_1



Man erkennt, dass das Aufeinanderfolgen der Punkte CD(Kosten 5) zu den teuersten Routen führt.