b) O espaço de solução para o **problema da mochila** consiste em 2^n maneiras distintas de escolher os itens de forma a maximizar a utilidade e minimizar o peso L. Outra maneira de verificar o custo exponencial é expressando o tamanho da entrada em termos do número de bits necessários para a representação binária dos inteiros que são parte da entrada. O peso p_i e a utilidade u_i podem ser expressos em termos de $x_i = \log p_i$ e $y_i = \log u_i$. Logo, $p_i = 2^x$ e $u_i = 2^y$, isto é, o peso e a utilidade são funções exponenciais do número de bits x e y utilizados para a entrada p_i e u_i. Logo, o algoritmo tem complexidade exponencial.

9.22.

a) O algoritmo mais eficiente conhecido é aquele que obtém todos os $(n-1)!$ caminhos e depois pega o maior deles.

b) $O(n!)$. São $(n-1)!$ caminhos com n adições em cada caminho. O problema é $\mathcal{NP}$-**completo**. Como não existe prova de que $P \neq \mathcal{NP}$ ou $P = \mathcal{NP}$, a resposta sobre se o algoritmo é ótimo (ou não) ainda não pode ser obtida.

c) O problema é $\mathcal{NP}$-**completo**. Um algoritmo não determinista polinomial é mostrado abaixo:

```
for i:= 2 to |v| do caminho[i] := escolhe(prox. vertice);
if |maior_caminho| >= k
then achou
else nao achou
```

Solução I: Transformar o problema em questão no **problema do caixeiro-viajante** clássico multiplicando cada distância por (-1) e obtendo $G' = (V, A^-)$. Neste caso, G' tem rota $<= (-k)$ se e somente se G tem rota $>= k$ que inclua todos os vértices. Logo, existe rota $>= k$ que inclua todos os vértices.

Solução II: Transformar o **ciclo de Hamilton** de $G = (V, A)$ para o problema do caixeiro-viajante máximo. (Ciclo de Hamilton é $\mathcal{NP}$-completo.) Como o grafo é completo, decidir se G tem um ciclo de Hamilton com comprimento $>= k$ (testando todos os ciclos hamiltonianos) é equivalente a resolver o problema em pauta.

Caracteres ASCII

Dec	Car	Dec	Car	Dec	Car	Dec	Car	Dec	Car	Dec	Car	Dec	Car
000	NUL	037	%	074	J	111	o	148	CCH	185	¹	222	Þ
001	SOH	038	&	075	K	112	p	149	MW	186	º	223	ß
002	STX	039	’	076	L	113	q	150	SPA	187	»	224	à
003	ETX	040	(	077	M	114	r	151	EPA	188	1/4	225	á
004	EOT	041	)	078	N	115	s	152	SOS	189	1/2	226	â
005	ENQ	042	*	079	O	116	t	153	SGCI	190	3/4	227	ã
006	ACK	043	+	080	P	117	u	154	SCI	191	¿	228	ä
007	BEL	044	,	081	Q	118	v	155	CSI	192	À	229	å
008	BS	045	-	082	R	119	w	156	ST	193	Á	230	æ
009	TAB	046	.	083	S	120	x	157	OSC	194	Â	231	ç
010	LF	047	/	084	T	121	y	158	PM	195	Ã	232	è
011	VT	048	0	085	U	122	z	159	APC	196	Ä	233	é
012	FF	049	1	086	V	123	{	160		197	Å	234	ê
013	CR	050	2	087	W	124	|	161	¡	198	Æ	235	ë
014	SO	051	3	088	X	125	}	162	¢	199	Ç	236	ì
015	SI	052	4	089	Y	126	~	163	£	200	È	237	í
016	DLE	053	5	090	Z	127	DEL	164	¤	201	É	238	î
017	DC1	054	6	091	[	128	PAD	165	¥	202	Ê	239	ï
018	DC2	055	7	092	\	129	HOP	166	¦	203	Ë	240	ð
019	DC3	056	8	093	]	130	BPH	167	§	204	Ì	241	ñ
020	DC4	057	9	094	^	131	NBH	168	¨	205	Í	242	ò
021	NACK	058	:	095	_	132	IND	169	©	206	Î	243	ó
022	SYN	059	;	096	‘	133	NEL	170	ª	207	Ï	244	ô
023	ETB	060	<	097	a	134	SSA	171	«	208	Ð	245	õ
024	CAN	061	=	098	b	135	ESA	172	¬	209	Ñ	246	ö
025	EM	062	>	099	c	136	HTS	173	–	210	Ò	247	÷
026	SUB	063	?	100	d	137	HTJ	174	®	211	Ó	248	ø
027	ESC	064	@	101	e	138	VTS	175	¯	212	Ô	249	ù
028	FS	065	A	102	f	139	PLD	176	°	213	Õ	250	ú
029	GS	066	B	103	g	140	PLU	177	±	214	Ö	251	û
030	RS	067	C	104	h	141	R1	178	²	215	×	252	ü
031	US	068	D	105	i	142	SS2	179	³	216	Ø	253	ý
032	˽	069	E	106	j	143	SS3	180	´	217	Ù	254	þ
033	!	070	F	107	k	144	DCS	181	µ	218	Ú	255	ÿ
034	"	071	G	108	l	145	PV1	182	¶	219	Û		
035	#	072	H	109	m	146	PV2	183	·	220	Ü		
036	$	073	I	110	n	147	STS	184	¸	221	Ý		